Зміст

Вступ

Метою дипломного проекту є розробка пристрою вводу дискретної інформації з використанням матричної клавіатури на базі мікроконтролера AT 90S8515.

Мікроконтролери сімейства AVR є пристроями синхронного типу. Дії, що виконуються в мікроконтролері, прив'язані до імпульсів тактового сигналу. Мікроконтролери мають повністю статичну структуру і можуть працювати при тактовій частоті від 0 Гц.

Робота деяких пристроїв мікроконтролера залежить від стану додаткових однобітових запам’ятовуючих елементів, - встановних бітів. Початкові значення встановних бітів записуються на заводі - виготівнику. Значення настановного біта може бути змінене тільки при програмуванні мікроконтролера.

В мікроконтролерах з самого початку у всіх комірках FlashROM записан код $FFFF, всіх комірках EEPROM – код $FF, біти захисту мають одиничне значення, а встановлюючі біти можуть мати різні значення. В трьох спеціальних комірках пам’яті записані сигнатурні байти, що визначають тип мікроконтролера.

В мікроконтролерах сімейства AVR існує два біти захисту – LB1 і LB2. При одиничному значенні обох бітів можливі запис і читання кодів.

1. Загальний розділ

1.1 Структура мікроконтролера

1.1.1. **Узагальнена структурна схема**

Мікроконтролери сімейства AVR мають єдину базову структуру. Узагальнена структурна схема мікроконтролера (МК) зображена

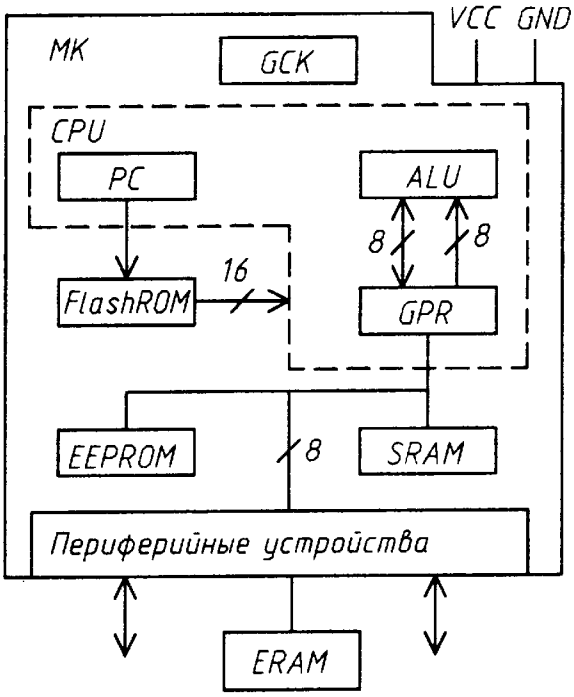

Рисунок 1. - Узагальнена структурна схема мікроконтролера

До складу мікроконтролера входять:

- генератор тактового сигналу (GCK);
- процесор (CPU);
- постійний запамятовуючий пристрій для зберігання програми виконаної за технологією Flash, (FlashRoom):
- оперативно запамятовуючий пристрій статистичного типу для зберігання даних (SRAM);

- постійний запамятовуючий пристрій для зберігання даних, виконаний за технологією EEPROM (EEPROM):
- набір периферійних пристроїв для введення і виведення даних і керуючих сигналів, і виконання інших функцій.

До складу процесора (CPU) входять:

- лічильник команд (PC);
- арифметико-логічний пристрій (ALU); .
- блок регістрів загального призначення (GPR).

Для нумерації регістрів вводу - виводу використовуються номери від 0 до 63 (від $00 до $3F; де $ - покажчик шістнадцятиричного коду). Кожному регістру привласнено ім'я, пов'язане з функцією, що виконує цей регістр. Мікроконтролери різних типів мають різний склад регістрів вводу - виводу, при цьому регістри з однаковими номерами можуть мати різні імена. Число регістрів вводу - виводу у мікроконтролерів різних типів вказане в таблиці. 1, в колонці IOR.

Робота деяких пристроїв мікроконтролера залежить від стану додаткових однобітових запамятовуючих елементів, - встановних бітів *(Pise Bits).* Початкові значення встановних бітів записуються на заводі-виготівнику. Значення настановного біта може бути змінене тільки при програмуванні мікроконтролера.

1.1.2. Генератор тактового сигналу

Мікроконтролери сімейства AVR є пристроями синхронного типу. Дії, що виконуються в мікроконтролері, прив'язані до імпульсів тактового сигналу. Мікроконтролери мають повністю статичну структуру і можуть працювати при тактовій частоті від 0 Гц.

Як генератор тактового сигналу (GCK) використовуються:

- внутрішній генератор із зовнішнім кварцевим або керамічним резонатором (XTAL);

- внутрішній RC-генератор (IRC);
- внутрішній генератор із зовнішнім RC-ланцюгом (ERC);
- зовнішній генератор (EXT).

Генератори тактового сигналу, використовувані в мікроконтролерах різних типів, вказані в таблиці. 1.

Таблиця 1. - Генератори тактового сигналу

Тип аск	Тип												
	t11	t12	t15	2323	2343	1200	2313	t28	4333	8515	8535	т163	т103
XTA	+	+		+		+	+	+	+	+	+	+	+
IAC	+	+	+		+	+		+				+	
EAC	+	+						+				+	
EXT	+	+		+	+	+	+	+	+	+	+	+	+

Зовнішній генератор (EXT) підключається до виведення XTAL1, при цьому виведення XT AL2 залишається не підключеним

У генераторах з RC-ланцюгом тактова частота визначається параметрами ланцюга, але змінюється в значних межах при зміні напруги живлення мікроконтролера.

1.1.3. Процесор

Процесор (CPU) формує адреси чергової команди, вибирає команду з пам'яті і організовує її виконання. Код команди має формат "слово" (16 біт) або "два слова".

До складу процесора окрім лічильника команд (PC), арифметико-логічного пристрою (ALU) і блока регістрів загального призначення (GPR) входять:

- регістр стану мікроконтролера SREG:
- регістр-показник стеку SP або SPL і SPH

У лічильнику команд адреси чергової команди формується шляхом добавлення 1 до числа, код якого зберігається в лічильнику команд. При пуску і перезапуску мікроконтролера в лічильник команд заноситися код числа 0 і перша команда вибирається з FLASHROM за адресою 0.

У арифметико-логічному пристрої (ALU) виконуються арифметичні і логічні операції. Операнди поступають з регістрів загального призначення (GPR). При виконанні одномісних операцій результат записується в регістр, з якого поступив операнд. При виконанні двомісних операцій результат записується в регістр, з якого поступив перший операнд.

Блок регістрів загального призначення (GPR) містить 32 восьмирозрядних регістра, яким привласнені імена RO, R1 ..., R31. У деяких операціях в ALU можуть брати участь лише регістри із старшими номерами (від R16 дО R31). Регістри з іменами від R24 дО R31 можуть здійснювати пари, що використовуються для зберігання слів, при цьому регістрі.

Регистр стану мікроконтроллера SREG містить 8 розрядів (SREG.7, SREG.6 .... , SREG.O).

1.1.4. Пристрій FlashROM, що запам'ятовує

Постійний запамятовуючий пристрій FLASHROM призначений для зберігання коду команд програми і констант. Осередок пам’яті містить 16 розрядів. В ній можуть зберігатися код команди формату "слово", половина коду команди формата "два слова" або коди двох констант.

При зчитуванні коду команд адреса в FLASHROM поступає із лічильника команд. При читанні констант адреса поступає з пари Z регістрів загального призначення.

Запис код в FLASHROM виконується в процесі програмування побайтно. У мікроконтролерах з великим числом виводів (20 і більше) байт може вводитися паралельно або послідовно. У мікроконтроллерах з малим числом виводів (8) байт вводиться послідовно.

Програмування може виконуватися з використанням додаткового джерела напруги + 12 В або без використання додаткового джерела. Послідовне програмування без використання додаткового джерела напруги проводиться з використанням трьох виводів мікроконтролера і може

виконуватися після установки мікроконтролера в апаратурі, де йому належить працювати (Downloading).

У мікроконтролері типу m 163 є можливість вводити і записувати в FLASHROM коди в процесі роботи мікроконтролера. Ця процедура виконується під управлінням спеціальної програми (Boot Loader), яка складається програмістом і записується в спеціальну секцію FLASHROM. При виконанні цієї програми використовується регістр SPMCR.

1.1.5. Запам’ятовуючий пристрій SRAM

Оперативно – запам’ятовуючий пристрій статичного типу SRAM призначений для зберігання даних, що отримуються в процесі роботи мікроконтролера. При виключенні напруги живлення мікроконтролера дані в SRAM втрачаються.

Адреса байту при зверненні до SRAM може бути вказаний в коді команди із зверненням до SRAM (пряма адресація) або заздалегідь записаний в пару регістрів X, У або Z (непряма адресація). Звернення до SRAM може виконуватися з використанням адреси, що зберігається в регістрі-покажчику стека.

Байт для запису в SRAM поступає з регістра загального призначення.

У адресний простір SRAM окрім адрес за якими виконується звернення до комірки пам'яті SRAM, включено 32 адреси для звернення до регістрів загального призначення (адреси від $00 до $1F) і 64 адреси для звернення до регістрів вводу-виводу (адреси від $20 до $5F).

1.1.6. Запам’ятовуючий пристрій EEPROM

Запам’ятовуючий пристрій EEPROM призначений для зберігання даних, записаних при програмуванні мікроконтролера і отримуваних в процесі виконання програми. При відключенні напруги живлення дані зберігаються.

При зверненні до EEPROM адрес записується в регістр адреси EEAR .У мікроконтролерах типу 8515, 8535 регістр адреси містить два восьмизарядні регістри - EEARL і EEARH. Байт призначений для запису, заноситься в регістр даних EEDR. Байт, що отримується при читанні, поступає в цей же регістр. Для управління процедурами запису і читання використовується регістр управління EECR.

Для запису байта в EEPROM необхідно:

1) записати адресу в регістр адреси;
2) записати байт в регістр даних;
3) встановити в одиничний стан розряд EEMWE регістра EECR
4) при EEMWE = 1 встановити в одиничний стан розряд EEWE регістра EECR.

Процедура запису виконується залежно від величини напруги живлення за 2,5-4 мс. При завершені запису розряд EEWE регістра EECR апаратно скидається в нульовий стан.

Розряд EEMWE зберігається в одиничному стані впродовж 4-х тактів після установки і апаратно скидається в нульове положення.

ДЛЯ зчитування байту з EEPROM необхідно:

1) записати адресу в регістр адреси;
2) встановити в одиничний стан розряд EERE регістру EECR.

Лічений байт поступає в регістр даних. Розряд EERE регістра EECR апаратно скидається в нульовий стан.

1.1.7. Набір периферійних пристроїв для вводу і виводу інформації

До периферійних пристроїв відносяться пристрої як:

- Паралельні порти вводу – виводу;
- Послідовний порт SPI;
- Послідовний порт UART;
- Послідовний порт TWSI;

- Таймери – лічильники загального призначення;
- Сторожовий таймер;
- Аналогово – цифровий перетворювач (АЦП);
- Аналоговий компаратор;
- Програмуючий апаратний модулятор;
- Блок переривання.

Паралельний порт вводу – виводу призначений для вводу и виводу даних. Мікроконтролери сімейства AVR мають від одного до шести портів. Порт може мати від трьох до восьми виводів.

Вивід порту може працювати в режимі входу або в режимі виходу. Направлення передачі біта установлюється для кожного вивода окремо.
Деякі виводи портів крім вводу і виводу бітів даних можуть використовуватись для виконання альтернативних функцій при роботі других пристроїв.

Послідовний порт вводу – виводу SPI призначений для вводу і виводу байтів при обміні даними з другими пристроями, які теж мають порт SPI. Обмін виконується під керуванням тактового сигналу порта. Пристрій який сприяє обміні і виробляє тактовий сигнал, являється ведучим. Пристрій виконуючий обмін при надходженні тактового сигналу являється відомим. В процесі обміну обидва пристрою послідовно біт за бітом одночасно видають і приймають байт. Обмін виконується з використанням трьох шин.
До одного ведучого пристрою можна підключати декілька відомих. Функції ведучого і відомого можуть змінюватись в процесі роботи системи.

Максимальна швидкість отримання і передачі дорівнює ¼ частоті тактового сигналу мікроконтролера.

Основним елементом порту SPI являється восьмирозрядний реверсивний здвигаючий регістр , який являється для паралельного ввода із шини даних мікроконтролера байта, призначеного для видачі: послідовної видачі байта на вихід мікроконтролера з одночасним вводом байта з входом

мікроконтролера і паралельної видачі введеного байта в буферний регістр для наступної передачі в шину даних мікроконтролера.

Порт SPI, який може працювати в режимі ведучого і ведомого і використовуватись для обміну даними в процесі роботи застосовується у мікроконтролерах типа 4433, 8515…

Аналогово – цифровий перетворювач (АЦП) формує десяти розрядний двоїчний код числа, пропорційно величині напруги живлення аналогового сигналу на вході мікроконтролера. В мікроконтролерах AVR до перетворювача можна підключати від чотирьох до восьми входів мікроконтролера.

Аналогово – цифровий перетворювач входить до складу периферійних пристроїв мікроконтролерів типа 4433, 8535.

Програмуючий апаратний модулятор призначений для формування імпульсного сигналу на виході для живлення світлодіодних індикаторів. Тривалість імпульсу задається програмним способом.

Сторожовий таймер призначений для знешкодження наслідків порушень в ході програми шляхом перезапуску мікроконтролера при виявлені порушень.

В склад сторожового таймера входять автономний генератор, перерахункова схема, регістр керування WDTCR і схема керування.

Генератор GWDT формує імпульсну послідовність. Частота проходження імпульсів залежить від напруги живлення мікроконтролера.

Для запуску сторожового таймера необхідно в ході програми виконати команду WDR і потім установити в одиничний стан розряд WDE регістра WDTCR.

Для установити сторожового таймера необхідно установити в одиничний стан розряд WDTOE регістра WDTCR і одночасно повторно установити в одиничний стан розряд WDE, потім через чотири такти після цього скинути в нульовий стан розряд WDE. Розряд WDTOE скидається в

нульовий стан апаратно через чотири такти після установки його в нульовий стан.

Перезапуск сторожового таймера виникає при виконані команди WDR в ході програми при одиничному стані розряду WDE.

Сторожовий таймер знаходиться у всіх типах мікроконтролерів.

1.2. Програмування мікроконтролера

1.2.1. Програмування мікроконтролера заключається в запису:

- ✓ Кодів команд програми і констант в FlashROM;
- ✓ Кодів вихідних даних в EERPOM;
- ✓ Потрібних значень встановлюючих бітів(Fuse Bits);
- ✓ Потрібних значень бітів захисту(Lock Bits);

В мікроконтролерах з самого початку у всіх комірках FlashROM записан код $FFFF, всіх комірках EEPROM – код $FF,біти захисту мають одиничне значення, а встановлюючи біти можуть мати різні значення. В трьох спеціальних комірках пам’яті записані сигнатурні байти, що визначають тип мікроконтролера.

В мікроконтролера сімейства AVR існує два біти захисту – LB1 і LB2. При одиничному значенні обох бітів можливі запис і читання кодів.

При перепрограмуванні мікроконтролеру необхідно завчасно «стерти» минулу інформацію.

При стиранні у всіх комірках FlashROM і EEPROM відтворюються коди $FFFF і $FF відповідно , біти захисту приймають одиничне значення, а встановлюючи біти і сигнатурні байти залишаються без змін. Допускається виконати до 1000 циклів «стирання-запису» для комірок FlashROM і до 100000 циклів для комірок EEPROM.

Запис і читання кодів при програмуванні виконуються паралельно або послідовно.

В мікроконтролера сімейства AVR реалізуються такі способи програмування :

- Паралельне програмування з використанням додаткового джерела живлення +12В(High – Voltage Parallel Programming .HVPP);
- Послідовне програмування з використанням додаткового джерела живлення +12В(High – Voltage Serial Programming .HVSP);
- Послідовне програмування без використанням додаткового джерела живлення +12В(Long – Voltage Serial Programming .LVSP);
- Самопрограмування (Self- Programming,SLFP).

Використання названих способів програмування в мікроконтролерах різних типів відмічена знаком «+».

Таблиця 2. - Способи програмування мікро контролерів

Тип МК	Спосіб програмування				Виводи порту SPI		
	HVPP	HVSP	LVSP	SLFP	MOSI	MISO	SCK
T11		+					
T12		+	+		PB0	PB1	PB2
T15		+	+		PB0	PB1	PB2
2323		+	+		PB0	PB1	PB2
2343		+	+		PB0	PB1	PB2
1200	+		+		PB5	PB6	PB7
2313	+		+		PB5	PB6	PB7
IT28	+						
4433	+		+		PB3	PB4	PB5
8515	+		+		PB5	PB6	PB7
8535	+		+		PB5	PB6	PB7
M163	+		+	+	PB5	PB 6	PB7
M103	+		+		PB2	PB3	PB1

Ввод і вивід байтів при програмуванні виконуються з використанням трьох виводів MOSI,MISO,SCK.

При програмуванні за способом LVSP в мікроконтролері можуть виконувати наступні операції:

1) дозвіл LVSP*;*
2) стирання запису*;*
3) запис байта в FLASHROM*;*
4) запис байта в буферних ЗУ *;*
5) перепис з буферного ЗУ в FLASHROM*;*
6) читання байта з FLASHROM*;*
7) запис байта в EEPROM;
8) читання байта з EEPROM):
9) запис бітів захисту *;*
10) читання бптов захисту*;*
11) запис настановних бітів*;*
12) читання настановних бітів*;*
13) читання сигнатурного байта *;*

14) читання калібровочного байту.

1.2.2. Запуск і перезапуск мікроконтролера

Запуск мікроконтролера відбувається при подачі напруги живлення на його виводи УСС і RESET. Перезапуск в процесі роботи виконується при появі на виведенні RESET нульового рівня сигналу з послідуючим поверненням до високого рівня і при переповнюванні сторожового таймера.

При пуску і перезапуску набуває активного значення внутрішній сигнал, який зберігає це значеннння протягом певного інтервалу часу. При активному значенні сигналу всі регістри вводу-виводу вводяться в початковий стан і в лічильник команд (PC) записується код числа 0.

Після закінчення інтервалу мікроконтролер приступає до виконання програми, починаючи з команди, записаної в FlashROM по нульовому адрес

У початковому стані регістрів введення-виводу всі розряди регістрів знаходяться в нульовому стані.

Виключенням є:

- регістр стану USR або UCSRA (NQ $0B) в порту UART, в який заноситься код $20 (U D RE = 1);
- регістр управління UCR або UCSRB *(N2* $0A) В порту UART, в який заноситься код $02 (RXB8 = 1);
- регістр OCR1B. В таймері-лічильнику T/c1 мікроконтролера типу t15, в який заноситься код $FF;
- регістр SPDR в порту SPI, початковий стан якого є невизначеним;
- регістри EEAR, EEARL і EEARH в блоці управління EEPROM мікроконтролерів t12, t15, 2323, 4423, 4433, 8535 і m163, початковий стан яких є невизначеним;
- регістр PORTA мікроконтролеру типу t28, в який заноситься код $04.

1.2.3. Енергозберігаючі режими роботи

Мікроконтролери сімейства AVR можуть бути переведені в

енергозберігаючі режими роботи. У Мікроконтролерів всіх типів можливі два енергозберігаючі режими - режим холостого ходу і режим пониженого енергоспоживання. Мікроконтролери, що мають таймери-лічильники з режимом відліку реального часу (8535, ш163, т103), крім того, можуть бути переведені в режим енергозбереження. Мікроконтролери типу t15 і m163, що мають, Ці аналого-цифровий перетворювачі, крім того, можуть переводитися в режим придушення перешкод роботі АЦП.Перехід в енергозберігаючий режим відбувається в ході програми при виконанні команди з мнемокодом SLEEP, якщо розряд SE регистру Cl MCUCR встановлений в одиничне положення.

Вибір режиму в мікроконтролерах, які можуть знаходитися в двох енергозберігаючих режимах, визначається станом розряду SM регістру MCUCR. При SM = Про встановлюється режим холостого ходу, при SM = 1 - режим зниженого енергоспоживання.

У мікроконтролерах, які можуть знаходитися в трьох і чотири эвергозберігаючих режимах, вибір режиму визначається комбинацією станів розрядів SM1 і SMO регістра MCUCR .

Вихід з енергозберігаючого режиму відбувається під час вступу дозволеного запиту переривання або перезапуску мікроконтролера. При виході за запитом переривання виконується перериваюча программа і далі здійснюється перехід до команди, розташованої після команди SLEEP за якою відбувся перехід в даний режим.

У режимі холостого ходу процесор зупинений, периферійні пристрої продовжують працювати, коди в регістрах загального призначення, осередках SRAM і регістрах вводу-виводу, розташованих в пасивних периферійних пристроях, зберігаються незміненими. Струм споживання Icc у мікроконтролерів різних типів зменшується в 2-4 рази. Якщо аналоговий компаратор не використовується для виходу з режиму, розряд ACD в регістрі ACSR для зменшення струму має бути встановлений в одиничний стан.

У режимі зниженого енергоспоживання (PDM) зупинений процесор і генератор тактового сигналу. Периферійні пристрої, окрім сторожового таймера і блоку переривань, не працюють. Коди в регістрах загального призначення, осередках SRAM і регістрах вводу-виводу зберігаються незмінними. Струм споживання Icc у мікроконтролерів різних типів має величину від декількох одиниць до декількох десятків

мкА. Напруга живлення Vcc може бути понижене до величини 2 В.

Вихід з режиму можливий при перезапуску мікроконтролера і по дозволеному запиту переривання із зовнішнього джерела при представленні запиту рівнем сигналу. Сигнал повинен зберігати низький рівень протягом деякого часу, поки генератор тактового сигналу входить в сталий режим. Режим енергозбереження (PSM) в мікроконтролерах, таймер-лічильник, що мають, режим відліку реального часу, відрізняється від режиму зниженого енергозбереження можливістю збереження роботи даного таймера після переходу в режим і можливістю виходу з режиму за запитом переривання від цього таймера при його роботі в режимі рахунку реального часу.

У режимі придушення перешкод роботі АЦП (ANRM) процесор остановлен. Працюють тільки аналого-цифровий перетворювач, блок переривання і сторожовий таймер. Вихід із режиму відбувається за запитом переривання із зовнішнього джерела

при представленні запиту рівнем сигналу, за запитом переривання від АЦП і при перезапуску мікроконтролера. У мікроконтролері типу m163 вихід з режиму можливий також за запитом переривання при пізнанні портом TWSI власної адреси.

1.3 Робота в CODEVISION AVR

1.3.1 Редагування файлу

Відкритий раніше в області редагування або знов створений файл можна відредагувати у вікні редактора, використовуючи відповідні команди. Ці

команди можна викликати з рядка меню, а можна клацнути правою кнопкою миші у вікні редактора і скористатися випадним меню, яке збігається з меню Edit (Правка).

Фрагменти тексту також можна вибрати за допомогою миші. Для цього потрібно встановити курсор в потрібне місце і, не відпускаючи ліву кнопку миші, перетягнути курсор в кінець фрагменту, що виділяється, після чого відпустити кнопку миші.

Можна вибрати слово, двічі клацнувши по ньому мишею.

Виділяючи і перетягуючи за допомогою миші, можна переміщати фрагменти тексту. Якщо при перетяганні утримувати натиснутою клавішу <Ctrl>, то цей фрагмент тексту буде скопійований в нове місце.

У вікні редагування можна переміщатися, скориставшись колесом миші.

1.3.2. Робота з проектами

Проект групує початковий файлі налаштувує компілятори, які використовуються для побудови конкретної програми. Операції, можливі при роботі з проектом в CodeVisionAVR.

Щоб добавити файл до проекту слід натиснути кнопку Add (Добавити).

При цьому відкриється діалогове вікно Add File To Project (Добавити файл до проекту). В цьому вікні необхідно вибрати файл, що потрібен і натиснутина кнопку «Відкрити».

1.3.3. Конфігурація проекту

Діалогове вікно Configure Project (Конфігурація проекту) має три закладки: Files (Файли), З Cotpiler (Компілятор Сі), Мега-серій Make (Після побудови).

1.3.4. Компіляція проекту

Для компіляції відкритого проекту слід вибрати відповідну команду. При

цьому буде запущений компілятор Сі CodeVisionAVR, який проведе асемблерний файл з ім'ям поточного проекту і з розширенням .asm. Цей файл можна проглянути і при необхідності змінити, відкривши його в редакторові.

За допомогою CodeViSionAVR можна скомпілювати тільки проект. Окремий файл скомпілювати не можна.

Процес компіляції у будь-який момент можна перервати .
Після того, як компіляція буде завершена, відкриється вікно Information (Інформація) з результатами компіляції.

Далі виводиться інформація про кількість скомпільованих рядків (тут - 156), причому підсумовується кількість рядків всіх включених в проект файлах і у файлах використовуваних бібліотек.

У наступних рядках виводиться наступна інформація:

- кількість помилок (errors) і попереджень (warnings) при компіляції;
- Bit variabIes size -- розмір, займаний бітовими змінними;
- Data Stack area - область в адресному просторі, займана стеком даних;
- Data Stack size - розмір області стека даних, який задається при конфігурації проекту;
- Estimated Data Stack usage - оцінне використання стека даних;
- Global variabIes area - область в адресному просторі, займана глобальними змінними;
- Global variabIes size - розмір області глобальних змінних;
- Hardware Stack area - область в адресному просторі, займана апаратним стеком; Stack
- Hardware Stack size - розмір області апаратного стека;
- Heap size - розмір так званої купи - області пам'яті, яка
- використовується для функцій динамічного розподілу пам'яті;
- EEPROM usage - використання EEPROM.

Кількість можливих помилок компіляції і попереджень буде вказано у вікні Information (Інформація).

Якщо натиснути на найменування помилки (або попередження) у вікні Navigator (Навігатор) або у вікні Messages (Повідомлення), то в області редагування у відповідному файлі буде виділений рядок, що містить цю помилку (або попередження).

1.3.5. Побудова проекту

Для побудови відкритого проекту слід вибрати відповідну команду «Команда Project~ Make Проект ~ Побудувати». При цьому спочатку буде запущений компілятор Сі CodeVisionAVR, який проведе асемблерний файл з розширенням .asm. Цей файл можна проглянути і при необхідності змінити, відкривши його в редакторові.

Процес компіляції у будь-який момент можна перервати «Команда Project~ Stop Compilatioп Проект ~ Зупинити компіляці».

Якщо не було виявлено ніяких помилок, то буде запущений асемблер Atmel AVR Avrasм2, який проведе вихідні файли, тип яких визначений раніше при конфігурації проекту на закладці З Compiler (Компілятор Сі).

Після цього відкриється вікно Information (Інформація), показуючи результати побудови. Це вікно має три закладки: Compiler (Компілятор), AssembIer (Ассемблер) і Programmer (Програматор)

Інформація на закладці Compiler (Компілятор) практично повністю збігається з інформацією, яка видається після виконання команди Project ~ Compile (Проект ~ Компілювати). Єдина відмінність (виділено червоною рамкою) - це додаткова інформація про розмір скомпільованої програми (Program size) і відсоток використання Flash-памяті (Пам'яті програм).

Кількість можливих помилок компіляції і/або попереджень буде вказана у вікні Information (Інформація) на закладці Compiler (Компілятор), а самі помилки компіляції і/або попередження будуть показані у вікні Messages (Повідомлення), розташованому під областю редагування, і у вікні Navigator (Навігатор) точно так, як і при виконанні команди Project ~ Cотpile (Проект ~

Компілювати).

Інформація на закладці Assembler (Асемблер) відображає результати роботи асемблера.

У перших рядках виводиться інформація про файли, які створив (Creating) асемблер.

Далі виводиться інформація про файли, які відтранслювали ассемблером (AssembIing), а також які файли підключені (Including) до ассемблерному файлу.

Потім приводиться наступна інформація про використання програмної пам'яті (Program memory usage):

- скільки слів (words) займає власне код програми (Code);
- скільки слів (words) займають табличні константи (Constant (dw/db»;
- невживаний (Unused) об'єм пам'яті;
- скільки слів (words) пам'яті зайнято всього (Total).

На закінчення при водиться кількість помилок після закінчення трансляції (AssembIy complete with no errors).

Інформація на закладці Programmer (Програматор) відображає значення лічильника програмувань чіпа (Сирий Programming Counter). Хоча кількість перепрограмувань чіпів Atmel дуже велика, воно все-таки кінцеве, і іноді корисно знати, скільки разів ваш чіп був перепрограмований.

2. Проектно - розрахунковий розділ

2.

2.1. Доступ до регістрів вводу - виводу

Кожному порту в мікроконтролері AVR відповідають три, як мінімум регістра:

- DDRx - регістр направлення даних (x – літера відповідного порту, наприклад регістр DDRB – регістр направлення даних Порту B). Цей регістр визначає, як повинен бути сконфігурований вивід порту – як вхід або як вихід. Якщо в біт цього порту записати 1, то відповідний вивід порту буде працювати як вихід; якщо записати 0, то відповідний вивід порту буде працювати як вхід. Наприклад, якщо в регістрі DDRB записано число 01000011, це значить, що 0-й, 1-й і 6-й виводи Порту B сконфігуровані як виходи, а останні – як виходи.

- PORTx - вихідний регістр порту x (наприклад, PORTB - вихідний регістр Порту B). В біти цього регістру записують ті значення, які хочуть отримати на виводах порту, при тому, що вводи сконфгуровані як виходи.

Якщо вивід сконфігурований як вхід, то при записувані в потрібний біт регістру PORTx 1 цей вивід стає «входом з підтяжкою», тобто цей вивід через резистор приблизно 100...150 кОм, який знаходиться всередині мікроконтролера, підключається («підтягується») до напруги живлення мікроконтролера. Таким чином, при відсутності сигналу на цьому вході буде рівень логічної 1.

- PINx - виводи порту x (наприклад, виводи Порту B).

Цей регістр містить значення логічних рівнів, які до теперішнього часу є на фізичних виводах порту. Цей регістр доступний тільки для читання.

Для доступу до регістрів вводу – виводу мікроконтролера AVR за допомогою інструкцій Асемблера IN і OUT компілятор використовує ключеві слова sfrb і sfrw.

Приклад:

```
/* Доступ до регістрів вводу/виводу */
/* Визначимо SFR (Регістри  загального призначення) для чіпу AT90S8515 */
sfrb PINC=0xl3;          //   Так як регістр PINA 8-бітний, то
sfrw TCNTl=0x2c;              до нього здійснюється
                         //   8-бітний доступ за допомого sfrb (sfr byte)
                         //   0x13 – це адреса порту PINC
                         //   так як регістр TCNT1 16-бітний, то до нього
                         //   здійснюється 16-бітний доступ з
                         //   допомогою sfrw (sfr word)
                         //   0x2c – це адреса регістру TCNT1

/* Основна функція програми
void main(void)
{
   unsigned char x;      // Об’являємо символьну зміну x
   x=PINC;               //  Читання виводів Порту A
   TCNTl=0xF4A6;         // Записування в регістр TCNT1L
}
```

Адреси регістрів вводу/виводу визначені в наступних заготовочних файлах, розміщених в піддиректорії.\INC:

tinyl3.h	90s8534.h	mega603.h
tiny22.h	90s8535.h	mega64.h
tiny26.h	mega103.h	mega8.h
90s2313.h	mega128.h	mega8515.h
90s2323.h	mega16.h	mega8535.h
90s2333.h	mega161.h	43usb355.h
90s2343.h	mega162.h	76c711.h
90s4414.h	mega163.h	86rf401.h
90s4433.h	mega169.h	94k.h
90s4434.h	mega32.h	
90s8515.h	mega323.h	

На початку програми за допомогою #include можна підключити потрібний файл для того мікроконтролера, який використовується в даному проекті.

2.1.2 Побітовий доступ до регістрів вводу/виводу

Побітовий доступ до регістрів вводу/виводу виконується шляхом добавлення вибраних бітів після ім’я регістра.

Оскільки побітовий доступ до регістрів вводу/виводу виконується з використанням інструкцій CBI, SBI, SBIC і SBIS, адрес регістра повинен бути в діапазоні 0...1Fh для sfrb і в діапазоні 0...1Eh для sfrw.

Приклад:

```
/* Доступ до регістрів  вводу виводу */
/*Визначимо SFR (Регістри загального призначення) для чіпа AT90S2313
*/
sfrb PORTB=0xl8;          // Пояснюємо компілятору, по яким адресам
sfrb DDRB=0xl7;           // знаходяться  ті або інші регістри (порти).
sfrb PINB=0xl6;           // Так як всі використані регістри 8-бітні, то
                          до
                          // виконується 8-бітовий доступ за допомогою
                          // sfrb. Звичайно ці оператори
                          // розташовуються
int a, b, c;              // в заголовочних файлах
                          // Об'являємо цілі зміні a, b, c
/*Основна функція програми */
void main (void)
{
/* Задамо направлення виводів Порту B*/
    DDRB. 0=0;            // Задамо біти 0…3 Порту B як входи
    DDRB. 1=0;            //
    DDRB. 2=0;            //
    DDRB. 3=0;            //

    DDRB. 4=0;            // Задамо біти 4…7 Порту B як виходи
    DDRB. 5=0;            //
    DDRB. 6=0;            //
    DDRB. 7=0;            //
    /* Встановимо виходи порту B*/
    PORT B. 0=0;          // Встановимо біти 0 і 1 Порту B як
    PORT B. 1=0;          // прості входи

    PORT B. 2=1;          // встановимо  біти 2 і 3 Порта B як «входи з
    PORT B. 3=1;          // підтяжкою», тобто кожний з цих виводів через
                          // резистор приблизно 100..150 кОм буде
                          // з'єднаний з шиною живлення. Таким чином, у
                          // відсутності сигналу на цих виводах буде
                          // рівень логічної 1
    PORT B. 4=0;          // встановимо біти 4 і 5 Порту B як виходи, на
    PORT B. 5=0;          // яких встановлений рівень логічного 0
    PORT B. 6=1;          // встановимо біти  6 і 7 Порту B як виходи, на
    PORT B. 7=1;          // яких встановлений рівень логічної 1

    /*опитаємо виводи Порту B, тобто їх фізичні значення*/
    a =PIN B. 0;          // Змінної a привласнимо значення виводу 0
                          // Порту B, тобто значення сигналу, на виводі 0
    b =PIN B. 5;          // змінної b привласнимо значення виводу 5
                          // Порту B, тобто b = 0
```

```
    c =PIN B. 6;              // змінної с привласнимо значення виводу 6
                              // Порту В, тобто с = 1
    if (PINB.2)               // якщо на виводі 2 порту В логічна 1,
    {                         // то виконується [група операторів 1];
     [група операторів 1]
    }
    Else                      // якщо 0 – то [група операторів 2]
    {
     [група операторів 2]
    }
…
}
```

Для кращого читання програми, бітами регістру вводу/виводу можливо привласнити символічні імена за допомогою #define:

Приклад:

```
/* Присвоєння символічних імен бітами регістрів вводу /виводу */
/* Ця примітивна програма запалює світлодіод при натисканні кнопки.
   Чіп:  AT90S2313.
   Світлодіод, підключений катодом до виводу 0 Порту В, а анодом через
   резистор 470 Ом – до плюса джерела живлення. Кнопка одним виводом
   підключена до 1 Порту В, а іншим до мінусу джерела живлення (до
   «загального»).  Частота кварца МГц, хоча в даному випадку вона може
   бути якась інша*/
#include <90s2313.h>    // підключаємо файл, де визначені адреса регістрів
/* Присвоєння символічних імен */
#define output_led PORTB.O        // вихід на світлодіод
#define input_button PINB.l       // вхід кнопки
/* Основна функція програми */
void main (void)
{
    /* Задамо направлення виводів Порту В */
    DDRB. 0=1;                // задамо біт 0 Порту В як вихід
    DDRB. 1=0;                // задамо біт 1 Порту В як вхід
    /* Встановимо виводи Порту В */
    output_led = 1;           // використовуємо символічне ім’я біта PORT B.0,
                              // запишемо в нього 1, щоб встановити вивід 0
                              // Порту В як вихід, на якому встановлений рівень
                              // логічної 1, щоб світлодіод не горів

    PORT B. 1=1;              // встановимо біт 1 Порту В як  «входи з
                              // підтяжкою»,щоб при віджатій кнопці на цьому
                              // виводі була 1, а при нажаті кнопки рівень
                              // змінювався на 0

    /* Нескінченний цикл */
```

```
    while (1);                    // Оператор циклу
    {
       /* Тіло циклу while */
       if (input_button==0);   // В умові використовуємо символічне ім'я
                               // виводу  PIN B. 1. Якщо кнопка натиснена,
                               // тобто на виводі PIN B. 1 логічний 0, то
       output_led=0;           // то запалюємо світлодіод
    else                       // інакше, якщо кнопка віджата
      output_led=1;            // гасимо світлодіод
    }                          // дужка закриває цикл while  ( )
}                              // дужка закриває функцію main ( )
```

2.3. LCD-функії

LCD-функції призначені для полегшення користувачеві сполучення між програмами Сі і алфавітно-цифровими LCD-модулями з вбудованим чіпом Но44780 від Hitachi або еквівалентним.

Контроллер HD44780 потенційно може управляти двома рядками по 40 символів в кожній (для модулів чотирма рядками по 40 символів використовуються два однотипні контроллери).

За допомогою цих виводів LCD обмінюється інформацією з мікроконтролером, що управляє. Мікроконтролер AVR посилає в LCD команди, керівники режимами його роботи, і коди символів, що виводяться. У свою чергу LCD може посилати мікроконтролеру AVR по його запиту інформацію про свій стан і дані зі своїх внутрішніх блоків пам'яті.

Три виведення LCD призначено для подачі живлячої напруги (VSS, VDD) і напруги зсуву (УО), яка керує контрастністю дисплея.

Виводи DBO ... DB7 використовуються для організації мультиплексированной шини «Команди / дані». На виводи RS, RJW, E (або El, E2 для LCD 4x40) мікроконтролер AVR виставляє керуючі сигнали. За допомогою сигналу на лінії RS мікроконтролер повідомляє контроллер LCD про те, що саме передається по шині: команда або дані. Якщо RS = 0, адресується регістр команд, якщо RS = 1 - регістр даних.

Дані через регістр даних, залежно від поточного режиму, можуть прочитуватися у відеопам'ять (DDRAM) або в ОЗУ знакогенератора (CGRAM) за поточною адресою.

Інформація, що потрапляє в регістр команд, інтерпретується пристроєм виконання команд як послідовність, що управляє. Прочитання регістра команд повертає в семи молодших бітах поточне значення лічильника адреси (AC), а в старшому розряді - прапор зайнятості (BE).

Сигнал на лінії E (або El, E2) є стробом, що супроводжує сигнали на шині «Команди / дані). Запис інформації в LCD відбувається по спаду цього сигналу.

Потенціал на виведенні R / W, що керує, задає напрям передачі даних: запис в RAM LCD (R/W = O) або при зчитуванні (R / W = 1).

Для випадку, коли мікроконтролер має обмежену кількість ліній вводу/виводу, передбачений другий варіант підключення LCD з використанням 4-бітової шини «Команди / дані). При цьому кожен байт даних передається по лініях DB4 ... DB7 послідовно двома тетрадами, починаючи із старшої.

Контроллер LCD після прийому байта команди або байта даних вимагає деякого часу для обробки отриманої інформації, протягом якого мікроконтролер AVR не повинен виконувати нові передачі.

Для того, щоб визначити, коли контроллер LCD закінчить свої внутрішні операції, мікроконтролер AVR може опитувати BUSY - прапор (прапор зайнятості BE), який скинеться тільки тоді, коли контроллер LCD звільниться. Другий, простіший спосіб полягає в тому, що мікроконтролер, що керує, знаючи, скільки часу потрібний LCD на обробку тієї або іншої команди, виконує тимчасову затримку після кожної передачі інформації.

Відеопам'ять (DDRAM), що має загальний об'єм 80 Б, призначена для зберігання кодів символів, що відображаються на LCD. Відеопам'ять організована в два рядки по 40 символів в кожній. Ця прив'язка є жорсткою і не підлягає зміні, іншими словами, незалежно від того, скільки реальних рядків матиме кожен конкретний LCD-модуль, 80x1 або 20x4, адресація відеопам'яті завжди переводиться до двох рядків по 40 символів.

При записі або зчитуванні буфера даних звернення здійснюється до

осередку, на який в даний момент указує курсор. У двустрочних LCD перші 40 осередків буфера даних зазвичай відображаються на верхньому рядку дисплея, а другі 40 осередків - на нижней.

Окрім DDRAM, контроллер LCD містить ще один блок пам'яті - знакогенератор. Його «прошивка», тобто відповідність код зображення символів, зазвичай є в описі LCD. З допустимих для розміщення в DDRAM код символи з кодами 0x00 ... мають спеціальне призначення - це перевизначувані символи, графічне зображення яких може призначити сам користувач, помістивши відповідну інформацію в області CGRAM. Для програмування доступні 8 перевизначуваних символів в режимі з матрицею 5x7 крапок. Для кожного з восьми перепрограмованих символів CGRAM відводиться по 8 елементів пам'яті, кожнен з яких відповідає одному рядку крапок в зображенні символу. Для символу 0 (код символу 0x00) адреси елементів пам'яті - 0x00 ... 0x07, для символу 1 адреси елементів пам'яті - 0x08 ... 0x0F. Таким чином, перепрограмована частина знакогенератора містить 64 байти пам'яті (8x8).

Для кодування матриці використовуються горизонтально «укладені» байти, п'ять молодших бітів, які несуть інформацію про малюнок, 4-й біт кожного з 8 байтів матриці визначає ліву колонку символу, а 0-й - праву. Старші три біта не використовуються і можуть мати будь-які значення.

2.6. Проект Keypad

Цей проект демонструє роботу мікроконтролера з матричною клавиатурою. Мікроконтролер визначає натиснуту кнопку клавиатури і виводе її код на алфавітно- цифровий модуль LCD - модуль.

Цей проект містить один вихідний файл keypad.c. Текст цього файлу з русифікованими коментарями:

```
1:      /*
2:      Демонструється клавіатура 4x4
3:
4:      CodeVisionAVR C Compiler
5:      (C) 2000-2002 HP InfoTech S.R.L.
6:
7:
8:      Чіп: AT90S8515
```

```
9:          Підключаємо матричну клавіатуру
10:
11:         [Роз'єм PORTD STK500] [КНОПКИ] R1
12:         1 PD  -----0----1----2----3-----------------o+5B
13:                   |     |      |    |R2    |
14:         2 PD1 ----4----5----6----7----------------
15:                   |      |     |     |R3    |
16:         3 PD2 ----8----9----10---11---------------
17:                   |      |     |       |R4   |
18:         4 PD3 ----12---13---14---15--------------
19:               D1     |      |      |      |
20:         5 PD4 -|<|-       |      |     |
21:               D2               |      |      |
22:         6 PD5 -|<|------        |      |
23:               D3                    |      |
24:         7 PD6 -|<|-------------       | R1…R4=10k…47k
25:               D4                      |
26:         8 PD7 -|<|------------------- D1…D4=1N4148
27:
28:    Використовуємо цифро - буквений LCD  2x16, підключений до роз'єму
PORTC STK500 наступним чином
29:
30:
31:    [LCD] [Роз'єм PORTC STK500]
32:    1 GND – 9 GND
33:    2 +5V – 10 VCC
34:    3 VLC – керування контрастністю LCD напруги 0…1 в
35:    4 RS – 1 PC0
36:    5 RD – 2 PC1
37:    6 EN – 3 PC2
38:    11 D4 – 5 PC4
39:    12 D5 – 6 PC5
40:     13 D6 – 7 PC6
41:    14 D7 – 8 PC7
42:    */
43:    #asm
44:    .equ_1cd_port=0x15
45:    #endasm
46:
47:    #include <lcd.h>
48:    #include <stdio.h>
49:    #include <delay.h>
50:    #include <90s8515.h>
51:
52:    // частота кварцу [Гц]
```

```
53:  #define F_XTAL 3686400L
54:  // PORTD0…3 є входами рядків
55:  #define KEYIN PIND
56:  // PORTD4…7 є входами стовбців
57:  #define KEYOUT PORTD
58:  // використовуємо для ініціалізації рахунку таймера/лічильника 0.
59:  #define INIT_TIMER0 TCNT0=0x100L-F_XTAL/64L/500L
60:  #define FIRST_COLUMN 0x80
61:  #define LAST_COLUMN 0x10
62:
63:  typedef unsigned char byte;
64:
65:  // зберігаємо стан кожної кнопки як біт
66:  // біт 0 буде KEY0, біт 1 KEY1, …
67:  unsigned keys;
68:  // буфер LCD-дисплея
69:  char buf [33];
70:
71:  керування за переповненням таймера/лічильника 0 кожні 2 мс
72:  interrupt [TIM0_OVF] void timer0_int(void)
73:  {
74:  static byte key_pressed_counter=20;
75:  static byte key_released_counter, column=FIRST_COLUMN;
76:  static unsigned row_data, ctr_key;
77:
78:  //заново ініціалізуємо таймера/лічильник 0
79:  INIT_TIMER0;
80:  rom_data<<=4;
81:
82:  //отримаємо групу 4 кнопок в змінній rom_data
83:  rom_data|=~KEYIN&0Xf;
84:  column>>=1;
85:
86:  if (column ==0) goto new_key;
87:    {
88:          column=FIRST_COLUMN;
89:          if (rom_data==0) goto new_key;
90:          if (key_released_counter==9) crt_key=rom__data;
91:          else
92:            {
93:          if (--key_pressed_counter==9) crt_key=rom_data;
94:          else
95:            {
96:          if (rom_data!=crt_key)
97:            {
```

```
98:             new_key:
99:             key_pressed_counter=10;
100:            key_released_counter=0;
101:            goto end_key;
102:            };
103:        if (!key_pressed_counter)
104:            {
105:            keys=row_data;
106:            key_released_counter=20;
107:             };
108:         };
109:    };
110:  end_key: ;
111:  row_data=0;
112:  };
113:  //вибираємо наступний стовбець, входи будуть
114:  //з підтягуючи ми резисторами
115:  KEYOUT=~COLUMN;
116:  }
117:
118:  //перевіряємо правильність натиснення кнопки
119:  unsigned inkey (void)
120:  {
121:  unsigned k;
122:  if (k=keys) keys=0;
123:  return k;
124:  }
125:
126:  void init_keypad (void)
127:  {
128:  DDRD=0Xf0;
129:  INIT_TIMER0;
130:  TCCR0=3;
131:  TIMSK=2;
132:  #ASM (”sei”)
133:  }
134:
135:  main ()
136:  {
137:      unsigned k;
138:      init_keypad ();
139:     lcd_init (16);
140:     lcd_putsf (”CVAVR Keypad”);
141:
142:     //зчитуємо кнопку і відображаємо її код
```

```
143:    while (1)
144:    {
145:      lcd_gotoxy (0,1);
146:      if (k=inkey () )
147:      {
148:          sprint (buf, ”key code=%Xh”, k);
149:          lcd_puts (buf);
150:        }
151:        else lcd_putsf (”NO KEY ”);
152:        delay_ms (500);
153:    }
154: }
```

У цьому файлі знаками /* (початок) і * / (кінець) виділений блок з коментарями, а знаком / / - рядки з коментарями.

У блоці з коментарями (рядки 1 ...42) даємо рекомендації по підключенню матричної клавіатури і LCD-модуля до відладочной плати STK500 при апаратній реалізації даного проекту.

У рядках 43 .. .45 оголошені, який порт мікроконтролера буде використовуватися для зв'язку з модулем LCD. Для цього в програму включений ассемблерний код. Директива #asm (рядок 43) говорить компілятору про початок ассемблерного коду, адиректива#endasm (рядок 45) про його завершення (див. *ДІРЕКТІВЫ #asm і #endasm).* У рядку 44 асемблерна директива .equ привласнює ідентифікатору _lcd_port значення 0x15. Це значення, відповідне адресі регістра PORTx вибраного порту, знаходимо у файлі 90s8515.h у відповідному рядку: «sfrb PORTC=OxI5;».

При трансляції асемблерного коду, отриманого при компіляції даного проекту компілятором Ci CodeVisionAVR, асемблер замість цього ідентифікатора підставить його значення.

У разі модифікації програми при використанні іншого порту для підключення LCD-модуля достатньо буде лише замінити значення вказанної директиви .equ, не змінюючи решти тексту програми.

Директивами #include (рядки 47 ... 50) підключаються: LCD-функції - файл lcd.h *;* функції вводу/виводу - файл stdio.h*;* функції затримки - файл

delay.h і заголовочний файл для використовуваного мікроконтролера AVR (AT90S8515) - файл 90s8515.h. Перед компіляцією процесор компілятора вставить замість цих рядків текст відповідних файлів.

Директивами #define (рядки 52 ... 61) визначаються ідентифікатори F_XTAL, KEYIN, KEYOUl; INIT_TIMERO, FIRST_COLUMN і LAST_COLUMN. Після цих рядків перед компіляцією процесор компілятора замінить в тексті програми ці ідентифікатори на їх значення: F_XTAL на 3686400L, KEYIN на PIND і так далі. У разі модифікації програми для іншої частоти кварцу, для іншого порту підключення клавіатури і так далі достатньо буде лише замінити значення в указаних директивах #define, не змінюючи решти тексту програми.

У рядку 63 оголошується тип даних byte, який відповідає типу даних unsigned char. Запис byte коротший і наочніший.

У рядку 67 оголошується глобальна змінна keys типу unsigned int (беззнакове ціле). Ключове слово unsigned означає unsigned int. Глобальні змінні автоматично ініціалізуються із значенням 0. В кожному біті цієї змінної зберігатиметься стан відповідної кнопки клавіатури У біті 0 зберігатиметься стан кнопки КЕУ0, в біті 1 Кеу1 ..., у біті 15 – Кеу15. Значення 1 у відповідному біті відповідає натиснутій кнопці, значення 0 - відтиснутій. Тобто якщо натиснута кнопка Кеу6, то значення змінної keys винне побут:: 0b 0000 0000 0100 0000 = 0xo040.

У рядку 69 оголошується глобальний символьний масив (строкова змінна) buf, що складається з 33 членів. При цьому всі елементи цього масиву автоматично ініціалізувалися із значенням 0.У цьому масиві зберігатиметься інформація, призначена для виводу на LCD.

Оскільки в налаштуваннях проекту на закладці C Compiler (Компілятор Сі) вибрана опція char is unsigned, то всі дані типу char (байт) компілятор буде обробляти як дані типу unsigned char, тобто масив buf складатиметься з даних типу unsigned char.

Рядки 72..116 – цього вираження програми (функції) обслуговування

переривання по переповнюванню таймера/лічильника 0 timerO_int.

У рядку 72 здійснюється доступ до системи переривань мікроконтролера AVR, на що вказує ключове слово interrupt. Далі за за цим ключевим словом в квадратних дужках повинен йти номер вектора переривання. Але тут замість номера коштує ідентифікатор TIMO_OVF. Річ у тому, що цей ідентифікатор визначений директивою #define в заголовному файлі 90s8515.h. Перед компіляцією процесор компілятора замінить Timo_ovf на цифру 8 (номер вектора прерьшания), як визначено у файлі 90s8515.h. Замість TIMO_OVF в квадратних дужках можна було написати 8, але TIMO_OVF наочніше. Далі йде ключове слово void, що вказує на те, що функція не повертає ніяких значень. Потім ім'я функції timerO_int (може бути довільним, але краще із смисловим навантаженням). В звязку з цим іменем далі в програмі здійснюється виклик даної функції. Ключове слово void в дужках вказує на те, що в цю функцію не передаються ніякі параметри.

У рядку 73 відкриваюча фігурна дужка вказує початок тіла програми обслуговування переривання.

У рядках 74 ... 76 оголошуються локальні змінні різних типів із специфікатором класу пам'яті static (статичні). Такі змінні мають глобальний час життя і зону видимості усередині функції, в якій вони оголошені. При виконанні програми статичні змінні ініціалізувалися тільки один раз, навіть якщо функція, де вони ініціалізувалися, викликається багато разів. Тип byte був визначений раніше, в рядку 63. Визначення unsigned має на увазі тип unsigned int.

Деякі змінні відразу ініціалізувалися, тобто їм привласнюються певні початкові значення. Якщо статичні змінні спеціально не ініціалізовані, то при запуску програми вони автоматично встановлюються в 0.

У рядку 79 перед компіляцією процесор компілятора замість ідентифікатора INIT_TIMERO підставить вираження визначене директивою #define в рядку 59. Таким чином, в цьому рядку відбувається ініціалізація тай мера/счетчика Про, тобто в його регістр TCNTO записується початкове

значення, яке обчислюється по формулі TCNTO = 0xI00L-F_XTAL/64L/500L. У цій формулі перед компіляцією процесор замінить ідентифікатор F_XTAL на 3686400L (рядок 5З). Буква L в кінці чисел означає, що ці константи мають тип long integer (довге ціле).

Спочатку, згідно пріоритетам операцій, зліва направо виконуються операції ділення (знак операції /, пріоритет З). Після цього виконується операція віднімання (знак операції -, пріоритет 4). Всі обчислення ведуться з довгими цілими, щоб не втратити значущі цифри. На закінчення значення результату всіх операцій записується (привласнюється) в регістр Tcnto (знак операції /, пріоритет 14).

У рядку 80 здійснюється операція зрушення вліво з наданням (знак операції - «=) поточного значення змінній row_data. Значення зрушується вліво на 4 біта, тобто значення О-го біта перейде в 4-й біт I-го - в 5-ій ..., ll-го - в 15-Й. Молодші 4 біта (з О-го по З-й) заповнюються нулями. Таким чином, значення зміних row_data зрушується на один ниббл (півбайт, або 4 бита) вліво. Молодший ниббл заповнюється нулями. Набутого значення привласнюється змінній row_data. її значення стане рівним Ob xxxx xxxx xxxx 0000, де xxxxxxxxxxxx - значення три молодших нибблів змінної row_data до операції. Значення саме старшого ниббла змінної row_data після цієї операції пропадає.

Перед компіляцією процесор замінить ідентифікатора KEYIN на PIND. PIND - це значення, лічене з регістра PINx Порту 0, тобто значення логічних рівнів, які до теперішнього часу присутні на фізичних (тобто реальних) виводах Порту D.

Відповідно до пріоритетів операцій порядок обчислення вираження наступний. Спочатку зчитується поточне значення PIND. Після цього із цим значенням виконується операція побітового логічного заперечення. Після її виконання всі 1 в значенні PIND замінюються на 0, а 0 - на 1. Потім виконується операція побітного I (знак операції &, пріоритет 8), набутого значення з числом 0XC = 0b 0000 1111. При цьому кожний біт першого

операнда порівнюється з відповідним бітом другого операнда. Якщо обидва зрівнюємих біта I, то відповідний біт результату встановлюється в 1, інакше - в 0. В результаті виходить значення 0b 0000 УУУУ, де УУУУ - значення молодшого ниббла, тобто бітів 0 ... 3-Й, регістра PIND, в якому 0 замінений на 1, а 1 - на О.

Останньою здійснюється операція побитного АБО з наданням поточного значення зміній row_data (0b xxxx xxxx xxxx 0000) і результату попередніх операцій (0b 0000 УУУУ). При цьому кожний біт першого операнда порівнюється з відповідним бітом другого. Якщо будь-який (або обидва) з зрівнюємих бітів дорівнює 1, то відповідний біт результату встановлюється в 1, інакше - в 0. Набутого значення (0b xxxx xxxx xxxx УУУУ) привласнюється змінною row_data.

У рядку 84 здійснюється операція зрушення управо з наданням (знак операції »=) поточного значення зміної column. Значення зрушується управо на 1 біт, тобто значення 7-го біта переходить в 6-й біт, 6-го - в 5-ій ..., 1-го - в 0-й. Значення 0-го біта пропаде, а в 7-й біт записується 0. Отримане значення привласнюється змінній column.

У рядках 86 ... 112 знаходиться скорочений оператор if-else (без else).

У рядку 86 - початок цього оператора, на що вказує ключове слово if. Вираження в дужках (умова оператора if-else) обчислюється в наступному порядку. Спочатку значення LAST_COLUMN (процесор підставить замість нього число 0x10 = 0b 00010000, рядок 61) зрушується управо на 1 біт (знак операції »). Потім набутого значення (0x08 = 0b 00001000) порівнюється із значенням змінної column, отриманим в попередньому рядку програми. Якщо вони рівні вираження істинно, якщо немає - помилково.

Якщо вираження в дужках істинно (тобто значення вираження відмінно від нуля), то буде виконуватися група операторів (складений оператор), ув’язнена у фігурні дужки, що знаходяться в рядках 87 (відкриваюча дужка) і 112 (закриваюча дужка). Тобто буде виконуватися тіло оператора if-else. Після цього виконання програми продовжується з

рядка 115. Якщо вираження в дужках помилково (значення вираження дорівнює нулю), то виконання програми відразу переходить на рядок 115, тобто тіло оператора if-else виконуватися не буде.

У складеному операторові в рядку 88 зміній column привласнюється значення FIRST_COLUMN (процесор підставить замість нього число 0x80 = 0b 1000 0000, рядок 60).

У рядку 89 знаходиться скорочений оператор if-else (без else). Оскільки в його тілі всього один оператор, вся конструкція записана в один рядок, без фігурних дужок. В умові оператора if-else (у дужках) перевіряється поточне значеннязмінної row_data, і якщо воно рівне 0, то виконується оператор goto, який здійснює перехід виконання программи на мітку new_key (на рядок 98). Якщо значення змінної row_data не рівне 0 (рівність помилкова), виконується наступний оператор (рядок 90).

У рядку 90 також знаходиться оператор if-else. В умові цього оператора if-else (у дужках) перевіряється поточне значення зміної keYJeleased_counter, і якщо воно не рівне 0 (істина), то виконується операція декремента (знак операції - --), тобто значення змінний keYJeleased_counter зменшується на 1, і виконання прорами переходить на рядок 91. Якщо значення змінної keYJeleased_counter рівне 0, то виконання прорами відразу переходить на рядок 91.

У рядку 91 знаходиться ключове слово else оператора if-else, що означає початок групи операторів, які будуть виконуватися, якщо умова оператора if-else (рядок 90) помилкова. Складний оператор поміщений у фігурні дужки, що знаходяться в рядках 92 (відкриваюча дужка) і 109 (закриваюча дужка).

У рядку 93 в умові чергового оператора if-else перевіряється вираження, що стоїть в дужках. Це вираження виконується в наступному порядку. Спочатку значення змінної key--pressed_counter декрементируется (знак операції --), тобто зменшується на 1. Потім набутого значення порівнюється з числом 9. Якщо вони рівні вираження істинно, якщо немає -

помилково. Якщо вираження істинно, то змінною crt_key привласнюється поточне значення змінної row_data, і виконання програми переходить на рядок 109 (точніше, 111, оскільки в рядках 109, 110 немає виконуючих операторів). Якщо вираження, що стоїть в дужках, помилково, то виконання програми переходить на рядок 94.

У рядку 94 знаходиться ключове слово else оператора if-else, що означає початок групи операторів (склад ний оператор), які будуть виконуватися, якщо умова оператора if-else (рядок 93) помилкова. Складений оператор поміщений у фігурні дужки, що знаходяться в рядках 95 (відкриваюча дужка) і 108 (закриваюча дужка).

У рядку 96 в умові оператора if-else перевіряється вираження, що стоїть в дужках. Це вираження буде істинне, якщо значення змінної row_data не дорівнює значенню змінної crt_key (знак операції !=), Інакше вираження помилкове. Якщо вираження істинно, то виконується група операторів (складений оператор), ув'язнена у фігурні дужки, що знаходяться в рядках 97 (відкриваюча дужка) і 102 (закриваюча дужка). Якщо вираження, що стоїть в дужках, помилково, то виконання програми переходить на рядок 103.

У рядку 98 знаходиться мітка new_key. Вона використовуються тільки для вказівки місця, куди передається управління оператором goto, що знаходиться в рядку 89.

У рядку 99 змінною key--pressed_counter привласнюється значення 10. У рядку 100 змінною keYJeleased_counter привласнюється значення 0.

У рядку 101 оператора goto здійснює перехід виконання програми на мітку end_key (на рядок 110).

У рядку 103 в умові оператора if-else перевіряється вираження, що стоїть в дужках. Це вираження буде істинне (не НУЛЬ), якщо значення змінної key_pressed_counter рівне 0, оскільки операція логічного заперечення НЕ (знак операції !) виробляє значення 0, якщо операнд є істина, і значення 1, операнд є брехня (нуль). Інакше вираження помилково. Вираження в дужках (!keY""pressed_counter) в даному випадку можна замінити іншим:

(keY""pressed_counter == 0).

Якщо вираження в дужках істинно, то виконується група операторів (составной оператор), ув'язнена у фігурні дужки, що знаходяться в рядках 104 (відкриваюча дужка) і 107 (закриваюча дужка). Якщо вираження, що стоїть в дужках, помилкове, то виконання програми переходить на рядок 108 (точніше, 111, оскільки В рядках 108 ... 110 немає виконуючих операторів).

У рядку 105 глобальною змінною keys привласнюється поточне значення локальної змінної row_data.

У рядку 106 змінною keYJeleased_counter привласнюється значення 20.

У рядку 110 знаходиться мітка end_key. Вона використовуються тільки для вказівки місця, куди передається управління оператором goto, що знаходиться в рядку 101.

У рядку 111 змінною row_data привласнюється значення 0.

У рядку 115 перед компіляцією процесор замінить ідентифікатор KEYOUT на PORTD (рядок 57). Таким чином тому вираженні спочатку береться значення змінної column, в якому всі 0 замінюються на 1, а 1 - на 0 (операція побітового логічного заперечення, знак операції -), а потім отриманий результат записується в регістр PORTX Порту D . При цьому значенні змінної column не змінюється.

На цьому виконання програми (функції) обслуговування переривання по переповненню таймера/лічильника 0 закінчується.

Функція timerO_int здійснює опит клавиатури. Кожний вивід молодшого ниббла Порту D (PDO ... PD3) конфігурований як вхід і підключений до відповідного ряду клавіатури. Кожний вивід старшого ниббла Порту D (PD4 ... PD7) конфігурований як вихід і через ДІОД (уо I ... Уо4) підключений до відповідного стовпця клавиатури. Спочатку на всіх чотирьох виводах молодшого ниббла Порту D (Pdo ..• Pdз) присутній рівень логічної 1 завдяки підтягаючим резисторам R1 ... R4.

Алгоритм опиту наступний. На кожний старшого ниббла Порту D (PD4 ... PD7) послідовно подається рівень логічного 0, зчитується значення

молодшого ниббла Порта D і записується в молодший ниббл змінної row_data, причому попереднє значення змінної row_data заздалегідь зрушується на 1 ниббл вліво. Якщо в поточному стовпці (на котрий поданий 0) є натиснута кнопка, то на входах в біті, який відповідає ряду з натиснутою кнопкою, також з'явиться 0. Таким образом, в змінній row_data виходить код натиснутої кнопки.

Блок 1 Виконується тільки один раз при найпершому виклику функції timerO_int, оскільки всі локальні змінні цієї функції оголошені з класом пам'яті static. При подальших викликах функції timerO_int блок 1 ігнорується.

Всі решта блоків функції timerO_int виконуються (відповідно до алгоритму) при кожному виклику цієї функції.

У блоці 2 відбувається нова ініціалізація таймера/лічильника 0, тобто починається новий відлік таймера/лічильника 0 для ініціації наступного переривання (рядок 79 програми) по його переповнюванню.

У блоці З відбувається зрушення з наданням значення змінної row_data на 1 ниббл (4 бита) вліво (рядок 80 програми).

У блоці 4 зчитується поточне значення регістра PIND; у цьому значенні ві 1 замінюються на 0, а 0 - на 1, і набутого значення заноситься в молодший ниббл змінної row_data (рядок 83 програми).

У блоці 5 відбувається зрушення з наданням значення змінної column на 1 біт управо (рядок 84 програми). Якщо набуте значення не рівне вираженню lAST_COLUMN»1 (блок 6), то всі 1 в значенні змінної column замінюється на 0, а 0 - на 1, і набуте значення записується (привласнюється) в регістр PORTD (блок 18), і виконання функції timerO_int закінчується.

Після чотирьох викликів функції timerO_int клавіатура буде повністю опрошена, а в змінній row_data знаходитиметься код натиснутої кнопки клавіатури. Якщо жодна кнопка не натиснута, то в змінній row_data буде значення Ob 0000 000000000000. Поточне значення змінної row _data привласнюється зміній crCkey.

Потім повний цикл опиту клавіатури для визначення натиснутої кнопки (чьоттирьохкратне повторення ланцюжка 2~3~4~5~6~18) повторюється 10 разів (10 значення змінної keYJressed_counter в блоці 14). Після кожного разу порівнюються значення змінних row_data і crt_key (блок 13) і, якщо всі 10 разів вони були рівними, то значення змінної row_data

привласнюється змінній keys (блок 16).

Якщо значення змінних row_data і crCkey хоч раз опинилися не рівні, то повний цикл опиту клавиатури для визначення натиснуто й кнопки знову повторюється 10 разів (змінній key_pressed_counter в блоці 14 знову привласнюється значення 10). Таким чином, зменшуються помилки, зв'язані з брязкотом контактів, нечітким натисненням кнопки клавіатури.

Після того, як значення змінної row_data привласнене змінній keys, тобто визначена натиснута кнопка клавиатури, починається опит клавиатури для визначення моменту відпуску кнопки. Знову здійснюється повний цикл опиту клавиатури (чьотирьохкратне повторення ланцюжка 2~3~4~5~6~18). Кількість повних циклів задається значенням змінної key_released_counter в блоці 16. Після кожного циклу опиту перевіряється умова рівності значення змінної row_data нулю (блок 16), що є критерієм того, що всі кнопки відтиснуті (немає нажатих кнопок). Якщо умова виконується, то починається новий цикл опиту клавиатури для визначення натиснутої кнопки. Якщо немає, то опрос клавиатури для визначення моменту відпуску кнопки повторюється.

На цьому закінчимо опис функції timerO_int і повертаємося до опису іншої частини програми, що знаходиться у файлі keypad.c.

Рядки 119 ... 124 - це визначення функції inkey.

У рядку 119 здійснюється оголошення функції inkey. Визначника unsigned (мається на увазі unsigned int (беззнакове ціле), задає тип значення, яке повертає функція. Далі слідує ім'я функції inkey (може бути довільні) . По цьому імені далі в програмі здійснюється виклик даної функції. Ключове

слово void в дужках вказує на те, що в цю функцію не передаються ніякі параметри.

У рядку 120 відкриваюча фігурна дужка вказує на початок тіла функції inkey.

У рядку 121 оголошується локальна змінна до типу unsigned int (беззнакове ціле). Ключове слово unsigned в дужках означає unsigned int .

У рядку 122 знаходиться скорочений оператор if-else (без else). Оскільки в його тілі всього один оператор, вся конструкція записана в один рядок, без фігурних дужок. В умові оператора if-else (у дужках) локальної змінної до привласнюється значення глобальної змінної keys, і якщо воно не рівне 0 (істинно), то глобальною змінною keys привласнюється значення 0 і виконується наступний оператор (рядок 123). Якщо значення змінної до після привласнення рівно 0 (помилково), то одразу виконується наступний оператор (рядок 123).

У рядку 123 знаходиться оператор retum, який завершує виконання функції, в якій він заданий, і повертає управління у вищестоящу функцію. Значення вираження, що стоїть за оператором retum, повертається у викликаючу функцію як значення викликаючої функції.

У рядку 124 закриваюча фігурна дужка вказує кінець тіла функції inkey.

Таким чином, функція inkey повертає значення локальної змінної до, якою привласнено значення глобальної змінної keys. Значення глобальної змінної keys після виконання функції inkey у будь-якому випадку дорівнює 0.

Рядки 126 ... 133 - це визначення функції iniCkeypad. Ця функція проводе ініціалізацію периферійних пристроїв мікроконтролера.

У рядку 126 здійснюється оголошення функції iniCkeypad. На початку стоїть ключове слово void, що вказує на те, що ця функція не повертає ніяких значень. Далі слідує ім'я функції init_keypad (може бути довільним). За цим іменем далі в програмі здійснюється вивід даної функції. Ключове слово void в дужках вказує на те, що в цю функцію не передаються ніякі параметри.

У рядку 127 відкриваюча фігурна дужка вказує початок тіла функції iniCkeypad.

У рядку 128 в регістр DDRx Порту D записується число 0xfo (0xfo=ob 1111 0000), тим самим всі виводи старшого ниббла цього порту (біти Po4 ... Po7) робляться виводами, а всі виводи молодшого ниббла (Біти~ POO ... Po3) - входами. Докладніше про регістр DDRx.

У рядку 129 замість, ідентифікатора INIT_TIMERO процесор компілятора підставить вираження, яке визначене в рядку 59. Таким чином, в регістр TCNTO таймера/лічильника 0 буде записано початкове значення, розраховане по відповідній формулі (рядок 79). Саме із цього значення починатиметься рахунок таймера/лічильника 0.

У рядку 130 в регістр управління таймером/лічильном 0 Tccro записується число 3, тим самим коефіцієнт ділення попереднього дільника частоти таймера/лічильника 0 встановлюється рівним 64, тобто таймера/лічильника 0 тактуватиметься частотою в 64 рази менше системною (частоти- кварцу).

При таких параметрах переривання по переповнюванню таймера/лічильника 0 відбуватимуться приблизно кожні 2 мс.

У рядку 131 в регістр маски переривання від таймерів/лічильників TIMSK записується число 2, тим самим вирішується переривання по переповнюванню таймера/лічильника 0 (якщо встановлений глобальний дозвіл переривань).

Докладніше про відповідні регістри мікроконтролера від Atmel на мікроконтролер AT90S8515. Інструкція асемблера sei встановлює прапор глобального переривання 1 у регістрі статусу SREG мікроконтролера, тим самим вирішуючи глобальне переривання.

У рядку 133 закриваюча фігурна дужка вказує на кінець тіла функції init_keypad.

Рядки 135 ... 154 - це визначення основної функції програми.

Ця функція обов'язково має бути присутньою у всіх програмах. Саме із цієї функції, в якому би місці програми вона не знаходилась, починається виконання програми.

У рядку 135 здійснюється оголошення функції таіп. На початку опущено (але мається на увазі) ключове слово void, що вказує на те, що ця функція не повертає ніяких значень; потім слідує ім'я функції таіп. У дужках опущено ключове слово void, яке вказує на те, що в цю функцію не передаються ніякі параметри.

У рядку 136 відриваюча фігурна дужка вказує на початок тіла функції таіп.

У рядку 137 оголошується локальна змінна до типу unsigned int (беззнакове ціле). Ключове слово unsigned в Cl. Не слід плутати її з локальною змінною до, оголошеною у функції inkey. Не дивлячись на одне ім'я, ці змінні різні . Вони видно тільки в тих функціях, де вони оголошені.

У рядку 138 здійснюється виклик функції init_keypad, яка визначена в рядках 126 ... 133. Ця функція ініціалізувала периферійні пристрої мікроконтролера.

У рядку 139 здійснюється виклик функції Icd_init з фактичним параметром 16 (кількість стовпців у використовуваному модулі LCD). Ця функція ініціалізувала модуль LCD, очищає дисплей і встановлює позицію для виводу символу в ряд 0 стовпця 0. Курсор не відображається.

У рядку 140 здійснюється виклик функції Icd_putsf, яка відображає в поточній дисплейній позиції рядок «CVAVR Keypad».

Обидві ці функції визначені в підключеному файлі Icd.h (рядок 47).

У рядках 143 ... 153 реалізований безкінечний цикл за допомогою оператора while .

У рядку 143 - початок цього оператора, на що вказує ключове слово while. Поки умова цього оператора (у дужках) істинна, тобто значення вираження відмінно від нуля, буде виконуватися група операторів (складений оператор, тіло оператора while), ув’язнена у фігурні дужки, що

знаходяться в рядках 144 (відкриваюча дужка) і 153 (закриваюча дужка). Вираження в скобках 1 (можна також написати 55 або 16), а не 0, тобто завжди істинно. Отже, ця група операторів буде виконуватися нескінченно, поки не відбудеться переривання програми.

У рядку 145 здійснюється виклик функції Icd_gotoxy з фактичними параметрами (0, 1). Ця функція встановлює поточну позицію дисплея відповідно в стовпець 0 і ряд 1. Нумерація рядів і стовпців починається с 0. Ця функція визначена в підключеному файлі Icd.h (рядок 47).

У рядках 146 ... 152 знаходиться оператор if-else.

У рядку 146 - початок цього оператора, на що вказує ключове слово ic.

Порядок обчислення вираження в дужках (умова оператора if-else) наступний. Спочатку викликається функція inkey, яка визначена в рядках 119 ... 124. Значення, яке повертає функція inkey, привласнюється локальній змінній до. Якщо це значення рівне 0, то вираження в дужках буде помилкове. У всіх інших випадках вираження буде істинне.

Якщо вираження в дужках істинно, то буде виконується група операторів (складений оператор), ув’язнена у фігурні дужки, що знаходяться в рядках 147 (відриваюча дужка) і 150 (закриваюча дужка). Після цього виконання програми буде продовжене з рядка 152. Якщо вираження в дужках помилково, то буде виконуватися оператор, в рядку 151, що знаходиться.

У рядку 148 функція sprintf здійснює форматні перетворення і поміщає результати в рядок buf. Специфікація перетворення «Кеу code = %ХЬ.> означає, що спочатку буде виведена послідовність «Кеу code ='>, потім значення локальної змінної до в шістнадцятиричному вигляді символами верхнього регістра (символ перетворення 'X'). Потім буде виведен символ «h». Ця функція визначена в підключеному файлі stdiO.h (рядок 48).

У рядку 149 функція Icd_puts відображає в поточній дисплейній позиції вміст рядка buf, який був сформований в попередньому рядку програми.

У рядку 151 знаходиться ключове слово else оператора if-else, що означає, що цей рядок буде виконуватися, якщо умова оператора if-else (рядок 146) помилково. У цьому рядку здійснюється виклик функції lcd_puts, яка відображає в поточній дисплейній позиції послідовність «NO KEY».

Функції lcd_puts і lcd_putsf визначені в підключеному файлі lcd.h (рядок 47).

У рядку 152 функція delay_ms генерує затримку 500 мс

У рядку 154 закриваюча фігурна дужка вказує на кінець тіла функції таіп.

Програма працює таким чином. Після подачі живлення починає виконуватися функція таіп. Послідовно викликається функція iniCkeypad, яка ініціалізувала Порт 0, таймер/лічильник 0 і настроює систему переривань, функція lcd_init, яка ініціалізувала LCD-модуль, і функція lcd_putsf, яка у верхній рядок LCD-модуля виводе послідовність «CVAVR Keypad).

Після цього програма переходить в безкінечний цикл, в якому спочатку курсор LCD встановлюється в початкову позицію нижнього рядка, потім викликається функція inkey, яка повертає код натиснутих кнопок клавиатури, і цей код відображається на LCD після напису «Key code =,). В кінці виводиться символ «h», що показує код, який виводиться в шістнадцятиричному вигляді. Якщо жодна кнопка не натиснута, то виводиться послідовність «NO KEY». Після цього генерується тимчасова пауза в 500 мс, і після неї безкінечний цикл повторюється знову. Під час виконання функції таіп постійно, через кожних 2 мс, відбувається переривання по переповнюванню таймера/лічильника 0 і викликається функція timerO_int, яка проводить опит клавиатури і залежно від натиснутих кнопок клавиатури формує відповідний код.

Код натиснутих кнопок слід інтерпретувати таким чином. Младший (перший) біт коди відповідає стовпцю 1, старший (четвертий) - стовпцю 4. Якщо в стовпці натиснута кнопка в ряду А, то в цьому біті буде число 1, якщо в ряду В - число 2, З - 4, D - 8.

При натисненні в одному стовпці два і більш за кнопки буде виділена сума їх кодів.

3. Економічний розділ

3.1 Техніко-економічне обґрунтування доцільності розробки і впровадження програмного продукту

В економічному розділі дипломного проекту зроблено розрахунки собівартості і розробки впровадження програмного продукту.

Вихідні дані для розрахунку собівартості програмного продукту надано у таблиці 3.

Таблиця 3. – Вихідні дані для розрахунку собівартості програмного продукту

Показник	Позначення	Одиниця виміру	Значення показника	Примітка
Показники ОФ, які використовуються для виконання розробки				
Кількість комп'ютерів	I	Шт.	—	Відповідно до завдання
Вартість комп'ютера	$Ц_{оф1}$	Грн.	2500	Відповідно до завдання
Вартість комплектуючих	$Ц_{оф2}$	Грн.	774.4	Відповідно до завдання
Вартість програмного забезпечення	$Ц_{оф3}$	Грн.	600	Відповідно до завдання
Трудові показники				
Річний фонд робочого часу	$Ф_{річ.}$	Год.	2004	Відповідно до завдання
Місячний фонд робочого часу	$Ф_{міс.}$	Год.	176	Згідно з діючим законодавством
Основна з/пл. (посадовий оклад) спеціаліста, який встановлює програму	$З_{осн. сп.}$	Грн.	300	Відповідно до завдання
Додаткова з/пл. спеціаліста (у відсотках до основної з/пл.)	$К_{дод.}$	%	20	Відповідно до завдання
Розмір відрахувань на соціальні заходи	$К_{соц.}$	%	39,1	Згідно з діючим законодавством
Плановий фонд робочого часу, необхідного для виконання розробки	$Ф_{план.}$	Год.	65	Відповідно до завдання

Продовження таблиці 3.

Норми і тарифи				

Річна норма амортизаційних відрахувань на електротехнічне устаткування (у відсотках до вартості)	$Н_а$	%	25	Згідно з діючим законодавством
Норматив витрат на утримання та експлуатацію основних засобів АСУ (до їх вартості), на рік	$Н_{вуеу}$	%	5	Згідно з діючим законодавством
Норматив накладних витрат (у відсотках до фонду основної заробітної плати)	$Н_{накл.}$	%	2	Відповідно до завдання
Вартість 1 КВт/год електроенергії	$Ц_{ел.}$	Грн.	0,246	Відповідно до діючих тарифів
Коефіцієнт корисного використання робочого часу (у відсотках)	$К_{корвик.}$	%	70	Відповідно до завдання
Потужність електрообладнання, яке використовується при виконанні розробки	М	КВт./год.	0,2	Відповідно до завдання

Таблиця 4 - Таблиця комплектуючих

№ п/п	Назва комплектуючого	Одиниця виміру	Кількість	Вартість одиниці, грн..	Сума, грн.
1	Резистор МЛТ-0,25-82 кОм±5%	Шт.	5	0.4	2
2	Світлодіоди EIR-303C ИК-диод 3 мм	Шт.	4	0.6	2.4
3	Клавіатура	Шт.	1	20	20
4	Мікроконтролер	Шт.	1	70	70
5	Дисплей	Шт.	1	680	680
Всього					774.4

3.2. Розрахунок собівартості розробки і впровадження програмного продукту

Витрати на утримання і експлуатацію основних засобів обчислюємо за формулою:

$$B_{вуеу} = \frac{\sum Ц_{оф} \cdot Н_{вуеу}}{100},\ грн.$$

(3.1)

де:

- $Н_{вуеу}$ – норматив витрат на утримання та експлуатацію основних засобів;

$\sum Ц_{оф}$ – загальна вартість основних засобів, які використовуються для розробки програмного продукту.

- .

$$B_{вуеу}=\frac{(2500+774,4+600)*5}{100}=193,72,\ грн.$$

Враховуючи, що норматив витрат на утримання та експлуатацію основних засобів розраховано на рік (річний фонд робочого часу відповідно до завдання на 2010 рік складає 2004 години), обчислюємо розмір цих витрат на період виконання завдання:

$$B_{вуеу\ завд}=\frac{B_{вуеу}}{Ф_{річ}}\cdot Ф_{пл},\ грн.$$

$$B_{вуеу\ завд}=\frac{193,72}{2004}\cdot 65=6,28,\ грн.$$

Суму амортизації основних засобів та нематеріальних активів за період планового часу розробки програмного продукту, розраховуємо за допомогою формули:

$$З_{ам}=\frac{\sum Ц_{оф}\cdot Н_{а}\cdot К_{пит.}}{100},\ грн.$$

(3.2)

де:

- $\sum Ц_{оф}$ – загальна вартість основних засобів, які використовуються для розробки програмного продукту;
- $Н_{а}$ – річна норма амортизаційних відрахувань, %;
- $К_{пит.річ.}$ – питома вага планового фонду часу на створення програмного продукту у річному фонді робочого часу.

$$К_{пит.\ річ.}=\frac{Ф_{пл.}}{Ф_{річ.}}$$

$$К_{пит.\ річ.}=\frac{65}{2004}=0,03$$

$$З_{ам} = \frac{(2500 + 774.4 + 600) \cdot 25 \cdot 0,03}{100} = 29,06 \ грн.$$

Витрати за статтею «Електрична енергія» розраховуємо за формулою:

$$З_{ел.} = М \cdot Ф_{пл} \cdot К_{кор.вик.} \cdot Ц_{ел}, \ грн. \quad (3.3)$$

$$З_{ел.} = 0,2 \cdot 65 \cdot 0,7 \cdot 0,246 = 2,24, \ грн.$$

Втрати, пов'язані з оплатою праці спеціаліста по розробці програмного продукту, обчислюємо за формулами:

$$З_{осн.} = З_{осн.зп} \cdot К_{пит.міс.} \ грн.$$

(3.4)

де:

- $З_{осн.зп.}$ – посадовий оклад спеціаліста, який виконує розробку програмного продукту, грн.;

- $К_{пит.міс.}$ – питома вага планового часу розробки програмного продукту у місячному фонді робочого часу.

$$К_{пит.міс.} = \frac{Ф_{пл.}}{Ф_{міс.}}$$

$$К_{пит.міс.} = \frac{65}{176} = 0,369$$

$$З_{осн.} = 300 \cdot 0,369 = 110,7 \ грн.$$

Додаткова заробітна плата:

$$З_{дод} = \frac{З_{осн} \cdot К_{дод}}{100} \ грн.$$

(3.5)

$$З_{дод} = \frac{110,7 \cdot 20}{100} = 22,14 \ грн.$$

Розмір відрахувань на соціальні заходи обраховуємо, використовуючи вище наведені розрахунки:

$$З_{соц.} = \frac{(З_{осн} + З_{дод.}) \cdot 39,1}{100}, \ грн$$

(3.6)

$$З_{соц.} = \frac{(110,7 + 22,14) \cdot 39,1}{100} = 51,94, \ грн$$

Загальні витрати на оплату праці спеціаліста за період створення програмного продукту та соціальні відрахування становитимуть:

$$Ф_{опл.\,з\,відрах} = З_{осн.} + З_{дод.} + З_{соц.},\ грн.$$

(3.7)

$$Ф_{опл.\,з\,відрах} = 110,7 + 22,14 + 51,94 = 184,78,\ грн.$$

Накладні витрати обчислюємо, використовуючи формулу:

$$В_{накл.} = \frac{(В_{вуеу.} + З_{ам.} + З_{ел.} + Ф_{опл.\,з\,відрах.}) \cdot Н_{накл.}}{100}$$

(3.8)

$$В_{накл.} = \frac{(6,28 + 29,06 + 2,24 + 184,78) \cdot 2}{100} = 4,447$$

Питому вагу кожної із статей витрат на розробку програмного продукту розраховуємо за формулою:

$$П_{пит.} = \frac{З_{показ.}}{В_{повн.}} \cdot 100, \qquad (3.9)$$

$$П_{пит.} = \frac{З_{осн.зп.}}{В_{повн.}} \cdot 100,$$

$$П_{пит.} = \frac{110,7}{226.8} \cdot 100 = 46.8$$

$$П_{пит.дод.зпл.} = \frac{З_{дод.зпл.}}{В_{повн.}} \cdot 100,$$

$$П_{пит.дод.зпл.} = \frac{22,14}{226.8} \cdot 100 = 9.8$$

$$П_{пит.соц.} = \frac{З_{соц.}}{В_{повн.}} \cdot 100,$$

$$П_{пит.соц.} = \frac{51,94}{226.8} \cdot 100 = 22.9$$

$$П_{пит.вуе.} = \frac{В_{вуеу}}{В_{повн.}} \cdot 100,$$

$$П_{пит.вуе.} = \frac{6.28}{226.8} \cdot 100 = 2,76$$

$$П_{пит.ам.} = \frac{З_{ам.}}{В_{повн.}} \cdot 100,$$

$$П_{пит.ам.} = \frac{29,06}{226.8} \cdot 100 = 12.8$$

$$П_{пит.ел.} = \frac{З_{ел.}}{В_{повн.}} \cdot 100,$$

$$П_{пит.ел.} = \frac{2,24}{226.8} \cdot 100 = 0,98$$

$$П_{пит.накл.} = \frac{В_{накл.}}{В_{повн.}} \cdot 100,$$

$$П_{пит.накл.} = \frac{4,447}{226.8} \cdot 100 = 1,96$$

Результати розрахунків витрат на розробку програмного продукту представлено у таблиці 5

Таблиця 5. - Склад витрат на розробку програмного продукту

Статті витрат	Умовне позначення	Одиниці виміру	Значення показника	Питома вага показника у загальній сумі витрат, %
Витрати на утримання і експлуатацію основних засобів	$В_{вуеу}$	Грн.	6,28	2,76
Сума амортизації основних засобів	$З_{ам.}$	Грн.	29,06	12,8
Витрати за статтею «електрична енергія»	$З_{ел.}$	Грн.	2,24	0,98
Витрати на оплату праці, у тому числі:	$Ф_{опл.з\ відрах}$		184,78	
- основна з/пл.	$З_{осн.}$	Грн.	110,7	48,8
- додаткова з/пл.	$З_{дод.}$		22,14	9,8
- відрахування на соціальні заходи	$З_{соц.}$		51,94	22,9
Інші накладні витрати	$В_{накл.}$	Грн.	4,447	1,96
Всього, витрати на розробку проекту	$В_{повн.}$	Грн.	226,8	100

Загальна вартість проекту складатиме суму витрат на придбання комплектуючих та витрат на розробку і впровадження програмного продукту

226,8 грн.

Висновок: Вище наведені розрахунки показують, що розробка програмного продукту силами наявних спеціалістів на підприємстві економічно більш вигідна – її вартість становить, згідно із розрахунками 226,8 грн.

4. Охорона праці

4.1. Охорона праці України

Основні задачі охорони праці це – виявлення і визначення виробничої небезпеки, професійної шкідливості і розробка заходів направлених на їх попередження або зменшення з метою ліквідації нещасних випадків і професійних захворювань, аварій і пожеж, а також забезпечення нормальних умов роботи.

Закон України «Про охорону праці» прийнятий Верховною Радою України 21 листопаду 2002 року визначає основні положення реалізації конституційного права громадян, що стосуються, на охорону їх життя і здоров'я в процесі трудової діяльності.

Закон складається з дев'яти розділів і сорока чотирьох статі, таких як:

1. Загальні положення (стаття 1 – 4).
2. Гарантії прав на охорону праці (стаття 5 – 12).
3. Організація охорони праці (стаття 13 – 24).
4. Стимулювання охорони праці (стаття 25 – 26).
5. Нормативно – правові акти по охороні праці (стаття 27 – 30).
6. Державне управління охороною праці (стаття 31 – 37).
7. Державний нагляд і суспільний контроль за охороною праці (стаття 38 – 42).
8. Відповідальність за порушення законодавства пір охорону праці (стаття 43 – 44).

В процесі роботи метою розділу охорони праці є проведення аналізу небезпечних і шкідливих виробничих чинників і виключення травматизму, професійних захворювань, а також захисту навколишнього середовища.

4.2. Вимоги до будівлі і приміщення підприємства

Страхова Група ТАС. Знаходиться за адресою проспект Леніна буд. 38.

Будівля побудована з силікатної цеглини і має третій ступінь вогнестійкості. Виробниче приміщення з постійним знаходженням людей має природне бічне освітлення. Окрім цього, відповідно до СНиП 11-4-79, передбачено загальне штучне рівномірне робоче освітлення. Для штучного освітлення існують світильники ОД. Електропостачання будівлі відповідає СНиП 1.02.01-85.

Загальні вимоги до робочого місця оператора ЕОМ.

- Робочі місця операторів за дисплеями слід розміщувати в спеціально відведеному приміщенні, яке відповідає гігієнічним вимогам щодо площі, умов природного освітлення та вентиляції.

- Робоче місце складається з стола з розміщеним на ньому екраном, клавіатурою і підставкою під документи, крісла, підставки для ніг.

- Доцільне розміщення клавіатури окремо від екрана. Це забезпечує вибір оптимального положення, висоти та нахилу всіх складових обладнання робочого місця учня.

- Екран повинен знаходитись нижче рівня очей прямо, або з нахилом на оператора. Кут зору, при якому забезпечується оптимальне розміщення символів на екрані в межах 0,5. Екран повинен розміщуватись на відстані 40-90см від очей оператора. Оптимальна відстань при висоті символів 2,5мм - 50см; при висоті символів 3-4мм її можна збільшити до 80см.

-Робоче крісло повинно бути рухоме. Короткі підлокітники крісла повинні забезпечувати положення рук трохи вище стола.

- Освітленість робочих місць повинна бути в межах від 300 до 500 як в зоні розміщення документів і клавіатури.

- В робочій зоні відношення яскравості поверхонь не повинно перевищувати 3:1, а між робочою поверхнею столу та навколишніми поверхнями (стіл, обладнання і т.п.) 10:1.

- З метою виключення на екранах дисплеїв яскравих плям, на робочому місці не повинно бути яскравих (блискучих) предметів великих розмірів.

- Для уникнення засвіток екранів і зниження перепадів яскравості в полі зору при природньому освітленні, робочі столи учнів необхідно розміщувати далі від вікон і таким чином, щоб віконні шиби знаходились збоку від учнів, а природне світло падало з лівої сторони. Вікна повинні бути оснащені шторами, які розсіюють світло, або регульованими жалюзі.

- Рекомендується робота на клавіатурі дисплея сліпим десяти пальцевим методом. Це дозволяє знизити втомлюваність зору за рахунок його постійного переключання з документа на екран і клавіатуру та шкідливу дію блищання клавіатури.

4.3. Аналіз небезпечних і шкідливих виробничих факторів

Мікрокліматичні параметри виробничого середовища - це поєднання температури, відносної вологості і швидкості повітря. Ці параметри в значній мірі впливають на функціональну діяльність людини, його самопочуття, здоров'я, а також і на надійність роботи обчислювальної техніки.

Причому у виробничих умовах характерна сумарна дія мікрокліматичних параметрів.

Великий вплив на мікроклімат в приміщеннях підприємств надають джерела теплоти - це ЕОМ, прилади освітлення, обслуговуючий персонал, а також сонячна радіація.

Таблиця 4.1. - Оптимальні норми мікроклімату для приміщень з ЕОМ

Період року	Категорія робіт	Температура повітря °С не більш	Відносна вологість повітря %	Швидкість руху повітря м/с
Холодний	Легка - 1а	22 -24	40 - 60	0,1
	Легка - 1б	21 - 23	40 - 60	0,1
Теплий	Легка - 1а	23 - 25	40 - 60	0,1
	Легка - 1б	22 - 24	40 - 60	0,2

Штучне освітлення в приміщеннях експлуатації моніторів і ЕОМ повинне здійснюватися системою загального рівномірного освітлення. У виробничих і адміністративно-суспільних приміщеннях, у випадках переважної роботи з документами, допускається застосування комбінованого освітлення.

Загальне освітлення слід виконувати у вигляді суцільних або переривистих ліній світильників, розташованих збоку від робочих місць, паралельно лінії зору користувача. При периметральном розташуванні комп'ютерів лінії світильників повинні знаходитися ближче до переднього краю, зверненого до оператора.

Джерелами шуму на підприємствах є самі обчислювальні машини (вбудовані в стійки ЕОМ вентилятори, принтери і т.д.), центральна система вентиляції і кондиціонування повітря і інше устаткування.

У виробничих приміщеннях, в яких робота на ЕОМ є допоміжною, рівні шуму на робочих місцях не повинні перевищувати значень, встановлених для даних видів робіт санітарними нормами допустимих рівнів шуму на робочих місцях.

При виконанні основної роботи на ЕОМ рівень шуму на робочому місці не повинен перевищувати 50дБА.

Комп'ютер є електричним пристроєм з напругою живлення 220/380 В трифазній чотирьохдротяній мережі із заземленою нейтраллю.

В основному, роботи по монтажу мережі полягають в збірці закуплених компонентів мережі і їх підключенні до електромережі.

Для забезпечення електробезпеки при монтажі, наладці і роботі з мережею необхідно звернути особливу увагу на створення захисних заходів від попадання користувачів і обслуговуючого персоналу під напругу, для запобігання електротравматизму при роботі з мережею.

На робочому місці необхідна наявність занулення.

Електроживлення робочого місця повинне бути підключене через рубильник, встановлений в місці, зручному для швидкого відключення живлення робочого місця.

4.4. Заходи щодо поліпшення умов праці

Дивлячись на специфіку зорової роботи з ВДТ, першорядним завданням є забезпечення необхідних умов візуальної роботи користувачів ВДТ за рахунок розподілу фарб в полі бачення працюючого і максимального зменшення засліплення від прямої і обмеженої близькості, від постійної пульсації зображення на обчислювальній техніці і інших чинників, які заважають і усилюють стомлюваність. Необхідно забезпечити як кількість, так і якісні параметри зображення.

Необхідно забезпечити відповідну обробку інтер'єру, тому що виблискувати на екранах і сліпити працюючих можуть не тільки вікна, але і інші поверхні великої яскравості, у тому числі: стелі, стіни, поверхні столів, шафи та одяг персоналу. Тому все повинне мати низький коефіцієнт віддзеркалення.

Мікроклімат в приміщенні повинен забезпечуватися кількістю повітря робочої зони згідно ГОСТ 12.1.005–88.

4.5. Засоби індивідуального захисту

Для профілактики порушень і підтримці працездатності необхідно дотримуватися регламентованої перерви для відпочинку. В період роботи за дисплеєм на підприємстві передбачено, щоб через кожні 40-45 хвилин роботи проводиться трьох-п'ятихвилинна перерва для відпочинку. Сумарно тривалість роботи з ВДТ розбивається на дві частини: в першу і другу половину робочого дня.

В період трудового процесу значно зменшується загальна м'язова активність при локальній напрузі грон рук. Для зниження стомлюваності перерви для відпочинку супроводжуються гімнастичними вправами для підтримки загального м'язового тонусу, а також профілактики кістково - м'язових порушень в поперековому відділі хребта.

Виконання цих вправ значно зменшує стомлюваність і попереджає розвиток загальних і зорових порушень.

Частіше всього використовують захисні засоби, до яких належать фільтри, встановлені над екранами дисплеїв за допомогою спеціальних пристосувань. В Україні використовують захисні екрани трьох типів: сітчасті, плівкові і скляні. В деяких випадках використовують пластикові екрани.

Також використовують такі захисні засоби, як захисні окуляри, халат, тапочки. Захисні окуляри виготовляють із спеціального поляризованого скла, які виконують функції фільтру.

Білий халат і тапочки використовують як засоби санітарної гігієни.

4.6. Пожежна безпека

4.6.1. Характеристика приміщення по пожежній небезпеці

Відповідно до ОНТП 24-86 дане приміщення відноситься до категорії “В”, і до класу П-Ііа по ПУЕ. Виникнення пожеж в будівлях і спорудах, розповсюдження вогню в них залежить від того, з яких конструкцій і матеріалів їх виконали. Характеристика виробничої будівлі: стіни силікатної цеглини, перекриття із залізобетонних конструкцій, двері дерев’яні. Будівля належить до третього ступеня вогнестійкості. Підвищення вогнестійкості досягається шляхом облицьовування будівельних конструкцій керамічною плиткою. Дерев’яні елементи, що використовуються в конструкціях будівлі, просочені антипіринами і в самих небезпечних місцях покриті вогнезахисними фарбами.

Головні причини, які викликають пожежу: порушення технологічного режиму, несправність електроустаткуванні, не правильна експлуатація виробничих і побутових електроприладів, конструктивні недоліки.

4.6.2. Пожежна профілактика

Будівля, де знаходиться офіс, має вільний під’їзд з усіх боків. Доріжки асфальтовані і закільцьовані.

Горюче сміття зберігається на спеціально пристосованих ділянках. Горючі матеріали і тара знаходяться на відстані не менше 0,5 метрів від світильників і електрики, а від труб індивідуального опалювання не менше 5см.

Відповідно “Типових правил пожежної безпеки” приміщення обладнані пожежними попереджувачами типа РПЧ-Д-1/0-К. Автоматичні засоби виявлення пожежі необхідно постійно підтримувати в робочому стані.

Система охоронно-пожежної сигналізації виконана відповідно з СНиП 2.04.09-81.

4.6.3. Засоби і заходи гасіння пожежі

Відповідно СНиП2.04.02.-84 для зовнішньої пожежегасінні на території встановлені пожежні гідранти, які підключені до міської мережі пожежного водопостачання і забезпечують подачу води. В будівлі розташований пожежний щит, обладнаний відповідним інвентарем пожежегасіння. Під ним знаходиться ящик з піском місткістю 2м^3.

В будівлі пожежні крани встановлені в коридорах. Пожежні крани розташовані на висоті 1,35м від підлоги в найдоступніших місцях. Кожний пожежний кран укомплектований рукавом діаметром 50мм і завдовжки 20м.

Для внутрішнього пожежегасіння подача води проводиться від з’єднаного господарсько-питного водопроводу.

Внутрішні пожежні крани знаходяться в спеціально опломбованих і пронумерованих шафах і не рідше за один раз в шість місяців перевіряються пуском води.

Використовування води в приміщенні, де знаходиться ВДТ з причини небезпеки пошкодження або повного виходу з ладу дорогого електронного устаткування можливо у виняткових випадках. При цьому пристрій ВДТ необхідно захистити від попадання води, закривають брезентом або тканиною.

Для гасіння пожежі в початковій стадії його виникнення приміщення забезпечені вуглекислотними вогнегасниками типа ОУ-5 з розрахунку: один на 100м^2 площі, в коридорах – через кожні 2м, але не менше одного на приміщення. В приміщенні з ВДТ знаходиться вогнегасник ОУ-5, а також пінні та порошкові вогнегасники

4.7. Технічна естетика

Для створення найсприятливіших умов праці необхідно враховувати психофізичні особливості людини, а також загальну гігієнічну обстановку.

При плануванні робочого місця необхідно враховувати зручність розташування дисплея, клавіатури і інших пристроїв ВДТ, а також зони руху рук оператора. Найзручніше сидіння з виїмкою, яке відповідає формі стегон і нахил назад. Спинка стільця повинна бути вигнутої форми: довжина її 0,3м, ширина 0,11м, радіус вигину 0,3-0,35м. Необхідно брати до уваги дану антропометрію. Рухи працівника необхідно сконцентрувати так, щоб групи м'язів були навантажені рівномірно, а зайві рухи усунені.

Раціональне колірне оформлення приміщення направлено на поліпшення санітарно-гігієнічних умов праці, підвищення його продуктивності і безпеки. Забарвлення виробничих приміщень впливає на нервову систему людини, його настрій, сприйняття запаху, смаку, і, врешті решт, на продуктивність праці. По цьому такий важливий вибір кольору приміщення: червоний колір збільшує м'язову напругу, оранжевий – стимулює діяльність, жовтий стимулює зір і нервову систему, зелений заспокоює, голубий розслабляє м'язову напруженість, фіолетовий створює відчуття спокою (СН 181-70). В приміщенні операторів ВДТ повинні бути встановлені декоративні рослини, квіти, різні штучні композиції, які створюють комфортну робочу атмосферу і сприяють гарному настрою працюючих.

5. **Охорона навколишнього середовища**

Нинішню екологічну ситуацію в Україні можна охарактеризувати як кризову, що формувалася протягом тривалого періоду через нехтування об'єктивними законами розвитку і відтворення природно-ресурсного комплексу України. Відбувалися структурні деформації народного господарства, за яких перевага надавалася розвитку в Україні сировино видобувних, найбільш екологічно небезпечних галузей промисловості.

5.1. Перелік факторів, що негативно впливають на стан навколишнього середовища. Вплив комп'ютера.

Завдяки більш детальному вивченню цієї проблеми вченими США, Канади, Іспанії, Венеції, а також України установлено, що у користувачів комп'ютерів присутні такі порушення функцій, як вегетативна судинна естонія з схиленням до підвищеного кров'яного тиску, остеохондроз, остроентерологічні порушення.

Результати дослідження взаємодії моніторів на користувача, проведені в Україні, вказують на те, що при великому строку взаємодії винайшли зміни в іонній системі. Їх глибина пропорційна величині навантаженню з найбільш яскраво вираженими ефективними у працюючих з комп'ютером 140-160 годин в місяць.

Ще одна негативна сторона роботи комп'ютером інтенсивності окислення міді. Біологічний аналіз явища показав посилення інтенсивності 2-3 рази.

У користувачів ПК спостерігаються також збільшене порушення функціонування нервової, судинної системи, збільшення кількості нещасних

родів жінок і аномалії плодів. У більшості жінок, працюючих з комп'ютером під час вагітності, плід розвивається аномально, при цьому важливі пороки розвивання зафіксовані в головному мозку. Результати дослідження останніх років вказали, що випромінювання ПК та телевізорів супроводжуються торсіонної компонентної, несучої інформації про процес, відбуваючись в електроприладах. Більшість електроприладів генерують торсіонне поле, які в визначеному ступені агресивні для людей.

Велика напруга (27 кВ), на першому аноді електронно-променевої трубки (ЕПТ), обчислює зарядну поляризацію вакууму, в результаті чого перед монітором генерується ліве торсіонне поле, а позаду праве торсіонне поле. Організм людини представляє собою складну торсіонну систему одного індивідуального типу, характеризується винятково інформацією, яка означає здоров'я. Поляризація інформаційного простору з порціонним полем визиває зниження біохімічних процесів.

Випромінювання низької частоти негативно впливають на центральну нервову систему, викликаючи головні болі, запаморочення, нудоту, депресію, безсоння, відсутність апетиту, виникнення синдрому стресу, причому нервова система реагує, навіть, на короткі за тривалістю впливу щодо слабких полів частоти: зменшується гормональний стан організму, порушуються біоструми мозку. Все це відображається на процесах вивчення і запам'ятовування.

Низькочастотне електромагнітне поле може стати причиною захворювань, хвороб серцево-судинної системи та кишково-шлункового тракту, воно впливає на білі кров'яні тільця, що призводить до виникнення пухлин, у тому числі й злоякісних.

5.2. Заходи щодо зменшення впливу комп'ютерної техніки на організм людини

Комп’ютерна техніка широко використовується в усіх галузях людської діяльності. Людина, яка працює з комп’ютером, постійно перебуває під впливом небезпечних і шкідливих виробничих факторів: електромагнітних полів, інфрачервоного та іонізуючого випромінювань, шуму й вібрації, статистичної електрики.

При роботі з комп’ютером основне навантаження припадає на всі елементи зорового аналізатора. Робота за комп’ютером характеризується тим, що постійний напружений погляд на екран дисплея зменшує частоту моргання. При цьому погіршується зволоження поверхні очного яблука сльозовою рідиною, яка захищає рогівку ока від висихання, пилюки та інших забруднень.

Наслідком напруженої зорової роботи за комп’ютером може бути не лише порушення функції зору, але й виникнення головного болю, посилення нервово – психічного напруження, зниження працездатності.

Для зменшення ризику захворювань необхідно проводити комплекс заходів:

- Дотримуватися раціонального режиму праці;
- Створювати оптимальні умови для праці у виробничому приміщенні;
- Проводити контроль за конструкцією і функціонуванням комп’ютера;
- Проводити диспансерне медико – гігієнічне обслуговування з цілеспрямованим проведенням оздоровчих і профілактичних заходів.

5.3. Перспективні напрямки, що до зменшення негативного впливу на довкілля

Для забезпечення захисту і досягнення нормованих рівнів комп'ютерних випромінювань необхідно застосовувати екранні фільтри, локальні світлофільтри (засоби індивідуального захисту очей) та інші засоби захисту, що пройшли випробування в акредитованих лабораторіях і мають щорічний гігієнічний сертифікат.

Віконні прорізи в приміщеннях використання ВДТ (віконно – дверні технології) і ПЕОМ повинні бути надані регульованими пристроями типу: жалюзі, зовнішніх козирків та інше.

Схеми розміщення робочих місць із ВДТ і ПЕОМ повинні враховувати відстань між робочими столами з моніторами (у напрямку тилу поверхні даного монітора й екрана іншого монітора), а відстань між бічними поверхнями моніторів – не менш 102см.

Проблема захисту користувачів і негативного впливу на них моніторів, може бути в деякому ступені вирішена за допомогою пристроїв розробленого в Київському політехнічному інституті. В залежності екранізуючих матеріалів використовуючи прозоре скло з металізованими плівками. Таке скло послаблює магнітне поле в 100-1000 раз.

Находять використання також еластичні екрани зі спеціальних тканин, в структурі яких тонкі металеві нитки створюють сітку з комірками розміром 0,5х0,5 мм. Вони забезпечують значне послаблення потужності електромагнітного поля.

Екран монітора повинен знаходитися на відстані 600-700 мм, але не нижче 500мм з урахуванням алфавітно-цифрових знаків і символів.

При виконанні роботи на моніторах і ПЕОМ, де працюють інженерно-технічні працівники, що здійснюють лабораторний, аналітичний чи вимірювальний контроль, рівень шуму не повинен перевищувати 60 децибел.

На робочих місцях у приміщеннях для розміщення гучних агрегатів машин рівень шуму не повинен перевищувати 75 децибел.

Шумливе устаткування, рівні шуму якого перевищують нормоване, повинне знаходиться поза приміщенням з монітором і ПЕОМ. Знизити рівень шуму в приміщеннях з монітором і ПЕОМ можна використанням звукопоглинаючих матеріалів із максимальними коефіцієнтами звукопоглинання в області частот 63-8000 Гц для обробки приміщень, підтверджених спеціальними акустичними розрахунками.

Додатковим звукопоглинанням служать однотонний засув з щільної тканини, що гармоніює з фарбуванням стін і підвищенні в складку на відстань 15м від огородження. Ширина засуву повинна бути в два рази більше ширини вікна.

Комп'ютерні технології, будучи великим досягненням людства, можуть мати негативні наслідки для здоров'я людей. Для збереження здоров'я необхідне дотримання встановлених гігієнічних вимог до режимів праці і організації робочих місць. Вибір режиму залежить від таких чинників, як тривалість зміни, час доби, вид діяльності, тяжкість і напруженість праці, санітарно-гігієнічні умови на робочому місці.

6. **Інформаційна безпека**

6.1 Засоби захисту інформації

Засоби захисту інформації — це сукупність інженерно-технічних, електричних, електронних, оптичних і інших пристроїв і пристосувань, приладів і технічних систем, а також інших речових елементів, використовуваних для вирішення різних завдань по захисту інформації, зокрема попередження витоку і забезпечення безпеки інформації, що захищається.

В цілому засоби забезпечення захисту інформації в частині запобігання навмисним діям залежно від способу реалізації можна розділити на групи:

- Технічні (апаратні) засоби. Це різні за типом пристрої (механічні, електромеханічні, електронні і ін.), які апаратними засобами вирішують завдання захисту інформації. Вони або перешкоджають фізичному проникненню, або, якщо проникнення все ж таки відбулося, доступу до інформації, зокрема за допомогою її маскування. Першу частину завдання вирішують замки, грати на вікнах, вартуючи, захисна сигналізація і ін. Другу — генератори шуму, мережеві фільтри, скануючі радіоприймачі і безліч інших пристроїв, що «перекривають» потенційні канали просочування інформації або що дозволяють їх виявити. Переваги технічних засобів пов'язані з їх надійністю, незалежністю від суб'єктивних чинників, високою стійкістю до модифікації. Слабкі сторони — недостатня гнучкість, відносно великі об'єм і маса, висока вартість.
- Програмні засоби включають програми для ідентифікації користувачів, контролю доступу, шифрування інформації, видалення залишкової (робочою) інформації типу тимчасових файлів, тестового контролю системи захисту і ін. Переваги програмних засобів — універсальність, гнучкість, надійність, простота установки, здібність до модифікації і розвитку.

Недоліки — обмежена функціональність мережі, використання частини ресурсів файл-сервера і робочих станцій, висока чутливість до випадкових або навмисних змін, можлива залежність від типів комп'ютерів (їх апаратних засобів).

- Змішані апаратно-програмні засоби реалізують ті ж функції, що апаратні і програмні засоби окремо, і мають проміжні властивості.
- Організаційні засоби складаються з організаційно-технічних (підготовка приміщень з комп'ютерами, прокладка кабельної системи з урахуванням вимог обмеження доступу до неї і ін.) і організаційно-правових (національні законодавства і правила роботи, що встановлюються керівництвом конкретного підприємства). Переваги організаційних засобів полягають в тому, що вони дозволяють вирішувати безліч різнорідних проблем, прості в реалізації, швидко реагують на небажані дії в мережі, мають необмежені можливості модифікації і розвитку. Недоліки — висока залежність від суб'єктивних чинників, зокрема від загальної організації роботи в конкретному підрозділі.
- Вбудовані засоби захисту інформації
- Антивірусна програма (антивірус) — програма для виявлення комп'ютерних вірусів і лікування інфікованих файлів, а також для профілактики — запобігання зараженню файлів або операційної системи шкідливим кодом.

Програмні засоби захисту інформації

- Спеціалізовані програмні засоби захисту інформації від несанкціонованого доступу володіють в цілому кращими можливостями і характеристиками, чим вбудовані засоби. Окрім програм шифрування і криптографічних систем, існує багато інших доступних зовнішніх засобів захисту інформації. З найчастіше згадуваних рішень слід зазначити наступні дві системи, що дозволяють обмежити і контролювати інформаційні потоки.

- Міжмережеві екрани (також звані брандмауерами або файрволами — від йому. Brandmauer, англ. firewall — «протипожежна стіна»). Між локальною і глобальною мережами створюються спеціальні проміжні сервери, які інспектують і фільтрують весь трафік сетевого/транспортного рівнів, що проходить через них. Це дозволяє різко понизити загрозу несанкціонованого доступу ззовні в корпоративні мережі, але не усуває цю небезпеку повністю. Захищеніший різновид методу — це спосіб маскараду (masquerading), коли весь витікаючий з локальної мережі трафік посилається від імені firewall-сервера, роблячи локальну мережу практично невидимою.

До апаратних засобів захисту відносяться різні електронні, електронно-механічні, електронно-оптичні пристрої. До теперішнього часу розроблено значне число апаратних засобів різного призначення, проте найбільшого поширення набувають наступні:

- спеціальні регістри для зберігання реквізитів захисту: паролів, ідентифікуючих код, грифів або рівнів секретності;
- пристрої вимірювання індивідуальних характеристик людини (голосу, відбитків) з метою його ідентифікації;
- схеми переривання передачі інформації в лінії зв'язку з метою періодичної перевірки адреси видачі даних.
- пристрою для шифрування інформації (криптографічні методи).

Для захисту периметра інформаційної системи створюються:

- системи охоронної і пожежної сигналізації;
- системи цифрового відео спостереження;
- системи контролю і управління доступом (СЬКУД).

Захист інформації від її витоку технічними каналами зв'язку забезпечується наступними засобами і заходами:

- використанням екранованого кабелю і прокладка проводів і кабелів в екранованих конструкціях;
- установкою на лініях зв'язку високочастотних фільтрів;
- побудова екранованих приміщень («капсул»);

- використання екранованого устаткування;
- установка активних систем зашумления;
- створення контрольованих зон.

7. Заходи з енергозбереження

7.1. Впровадження енергозберігаючих технологій

У питанні диверсифікації джерел поставок енергоносіїв Україна протягом останніх років зазнає поразку за поразкою. Доводиться констатувати — зовнішній напрям енергетичної політики держави далеко не найкращий. Зрозуміло, що у великій геоенергетичній політиці світових лідерів Україна навряд чи матиме великі здобутки — не втратити б того, що маємо. Але й на внутрішньому напрямі, себто у сфері реалізації політики енергозбереження, ми продовжуємо пасти задніх, підміняючи реальні кроки багаторічною риторикою.

Для України, економіка якої на порозі неабиякого стресу — ціна на газ у 2010 році може сягнути 300 доларів за тисячу кубометрів, — енергозбереження стає не просто елементом економічної доцільності, а питанням виживання. Уряд має грати на випередження ситуації, вдаючись до відповідних заходів уже сьогодні.

Споживання природного газу в Україні 2007 року порівняно з 2006-м зменшилося на 4,1 млрд. кубометрів. Однак наша держава стабільно посідає місце в першій десятці країн світу за обсягом споживання природного газу і третє місце — за обсягами його імпорту. Україна закуповує 50—55 млрд. кубометрів газу на рік, або 71% від його загальних обсягів споживання. Тобто частка імпорту в газовому балансі України критично висока.

При цьому держава витрачає шалені кошти на закупівлю газу. Якщо в 2007 році газ коштував Україні 6,5 млрд. дол. (ціна тисячі кубометрів газу становила 130 доларів), то в 2008-му — вже понад 9 млрд. дол. (179,5 дол. за тисячу кубометрів). Залежність від одного джерела газопоставок порушує баланс енергетичної безпеки і ставить економіку України в надмірну залежність від тенденцій зовнішньоекономічної політики Росії.

Одним з чинників катастрофічного стану паливно-енергетичного комплексу є надзвичайно велике споживання енергоносіїв на одиницю виробництва внутрішнього валового продукту: Україна витрачає в 2,6 разу більше, ніж країни Західної Європи і світу. Фактично Україна зажила слави однієї з найбільш енерговитратних країн світу: рівень витрат становить 2,2% від світової частки первинної енергії, тоді як кількість населення ледь сягає 1% від світової.

7.2. Енергозберігаючі технології у домашньому ПК

Потужність домашнього ПК зазвичай не перевищує 1 кВт, у глобальному масштабі це дає досить помітний ефект. У глобальному - навіть для офісу, у якому встановлено, наприклад, сто комп'ютерів, вартість електрики може помітно відбиватися на витратах фірми. Не кажучи вже про сервера та потужність робочої станції . Філософія енергозберігаючих технологій для ПК дуже проста - нічого не має працювати вхолосту, коли воно непотрібне. Вкрай важливим питання енергоспоживання є для ноутбуків: адже режим живлення впливає на час автономної роботи.

7.3. Процесор

Енергозберігаючі процесори від Intel та AMD - дуже популярна зараз тема. Від звичайних процесорів вони відрізняються тим, що можуть змінювати робочу частоту залежно від навантаження на систему (тобто, знаходиться система у режимі простою або запущено якусь ресурсоємку програму). Використання таких процесорів допомагає знизити рівень енергоспоживання приблизно у 1,5 рази.

7.4. Графічна система

З точки зору енергоспоживання, гарна відеокарта - це справжнісінька проблема. Сучасні потужні відеокарти споживають енергії не менше, ніж усі

інші елементи комп’ютера разом - від 110 до 270 Вт при повному навантаженні. Тобто перед виробниками зараз стоїть завдання - розробити відеокарту, яка могла б регулювати рівень споживання енергії залежно від режиму роботи ПК.

7.5. Накопичувач даних

В широкому використанні зараз знаходяться два види накопичувачів: магнітний Hard Disk Drive (HDD) та елементи флеш-пам’яті Solid State Drive (SSD), які, на відміну від перших, не мають рухомих елементів. SSD найчастіше використовуються у ноутбуках - саме через низький рівень споживання та компактні розміри. Щоправда, коштують вони недешево (близько 8 у.о./Гб) та ємність їхня ще досить обмежена порівняно із HDD. Натомість спеціалісти стверджують, що зниження енергоспоживання буде залежати від характеру роботи, а саме - частоти звернень до жорсткого диску. Тобто, виконуючі звичайні офісні задачі.

Висновки

Результатом виконання дипломного проекту є розробка і створення технологічної клавіатури на базі мікроконтролера сімейства AVR.

В загальному розділі розглянуто структура мікроконтролера, робота мікро контролера та робота в CodeVision AVR.

В проектно-розрахунковому розділі наведено доступ до регістрів вводу – виводу, наведені LCD функції мікро контролеру AT 8515 та проект «Keypad», вибір друкованої плати.

В економічному розділі: наведений розрахунок собівартості розробки.

В розділі “Охорона праці” описані рішення та заходи з поліпшенням умов праці, ефективність запропонованих заходів охорони праці.

В охороні навколишнього середовища описано перелік факторів, що негативно впливають на стан навколишнього середовища. Вплив комп’ютера на організм людини.

Розділ «Заходи з енергозбереження» містить інформацію про положення закону України «Про енергозбереження», в ньому описано енергозберігаючі технології у домашньому ПК.

Результати дипломного проекту можуть бути використані на інших промислових підприємствах. Що дозволяє раціонне використовувати робочий час.

Перелік посилань

1. Примак Т.О. Економіка підприємства. Навчальний посібник. - К. “ВІКАР”, 2001, - 178с.
2. Покропівний С.Ф. Економіка підприємства: Підручник. 2-ге вид. - К.:КНЕУ, 2000. - 528с.
3. Примак Т.О. Економіка підприємств. - К.: МАУП, 1999. - 108с.
4. Бойчик І.М., Харів П.С., Хопчан М.І. Економіка підприємств. Навчальний посібник. - Львів: “Сподом”, 2000. - 212с.
5. Гандзюк М.П., Желібо Є.П., Халімовський М.О. Основи охорони праці: Підручник для студентів вищих навчальних закладів. За редакцією М.П.Гандзюка. - К.: Каравела, 2003. - 408с.
6. Закон України “Про охорону праці” від 21 листопада 2002р.
7. Житецький В.Ц. Основи охорони праці. —Львів: Афіша, 2002. - 320с.
8. Геврик Є.О. Охороно праці.- К.:Ельга; Ніка-Центр, 2003. - 280с.
9. Лебедев М.Б. CodeVisionAVR: посібник для починаючих. – М.: Додека – XXI, 2008. – 592с.
10. Вірт Н. Алгоритми і структури даних / Переклад з англійського. - М.: Мир, 1989. - 360с.
11. Грінзоу Лу. Філософія програмування для Windows 95/NT / Переклад з англійського. - СПб.: Символ-плюс, 1997. - 640с.
12. Гук М. Інтерфейси ПК. Довідник. - СПб.: ПітерКом, 1999.Спірідонов В.І., Войтков В.Г. Обчислювальна техніка і програмування. – Хмельницький: ХТІ, 1992.